on

UNIT 1 INTRODUCTION TO NETWORK ADMINISTRATION

Structure **Page Nos.**

1.0 Introduction 5

1.1 Objectives 5

1.2 Roles and Responsibilities of Network Administrator 6

1.3 Linux and TCP/IP Internetworking Concepts 6

1.4 Using Network Clients 10

1.5 Understanding System Initialization 11

1.6 User Remote Administration Services and Tools 16

1.7 Summary 17

1.8 Answers to Check Your Progress 18

1.9 Further Readings 19

1.0 INTRODUCTION

Computer network is a telecommunications network that connects a collection of computers to allow communication and data exchange between systems, software applications, and users. The computers that are involved in the network that originate, route and terminate the data are called nodes. The interconnection of computers is accomplished with a combination of cable or wireless media and networking hardware. Two devices are said to be networked when a process in one device is able to exchange information with a process in another device. Networks may be classified by various characteristics, such as the media used to transmit signals, the communications protocols used to organize network traffic, network scale, network topology and organizational scope. The best-known computer network is the Internet.

Communication protocols define the rules and data formats for exchanging information in a computer network. Well-known communications protocols include Ethernet, a hardware and link layer standard that is widely used for local area networks, and the Internet protocol suite (TCP/IP), which defines a set of protocols for communication between multiple networks, for host-to-host data transfer, and for application-specific data transmission formats. Protocols provide the basis for network programming.

1.1 OBJECTIVES

After going through this unit, you will be able to:

- know the roles and responsibilities of a Network Administrator;
- know about network client and its purpose;
- understand LINUX system initialization; and
- understand remote system administration and available tools.

1.2 ROLES AND RESPONSIBILITIES OF NETWORK ADMINISTRATOR

A Network Administrator is an individual, who is responsible for configuring, commissioning and maintenance of network infrastructure and services. It also includes the computer hardware and software systems that make up a data network. In an organization, Network Administrator generally don't typically get involved directly with users, instead focus upon configuring, monitoring and maintenance of network components within organization's LAN/WAN infrastructure. Depending on the organization and its size, the Network Administrator may also involve in design and deployment of computer networks.

Roles of a Network Administrator

The roles of a Network Administrator include activities and tasks to be performed such as configuring, commissioning and maintenance of various network devices-routers, switches, VPN gateways, security devices-Firewall and IDS/IPS , creation of De Militarized Zones (DMZ) , IP addresses allocation & management. It also includes configuring and commissioning of various network services/protocols-DHCP, DNS, FTP, HTTP, NFS, etc.

Apart from roles, the Network Administrator is also responsible to

- Ensure data network connectivity
- Network monitoring and management
- Testing the network for breaches, if any
- Keeping an eye out for needed updates
- Update Access Control Lists (ACLs) time to time to regulate network traffic
- Security controls enforcement
- Preparing and implementation of security policy and standards

1.3 LINUX AND TCP/IP INTERNETWORKING CONCEPTS

Linux is a Unix-like computer operating system assembled under the model of free and open source software development and distribution. Linux was originally developed as a free operating system for Intel x86-based personal computers. It is a leading operating system and supports different computer hardware platforms like other operating systems. Linux also runs on embedded systems (a devices where the operating system is typically built into the firmware and highly tailored to the system) such as mobile phones, tablet computers, network routers, building automation controls, televisions, video game consoles and. The Android system, which is in wide use on mobile devices is built on the Linux kernel.

Typically Linux is packaged in a format known as a Linux distribution for desktop and server use. Some popular mainstream Linux distributions include Debian (and its derivatives such as Ubuntu and Linux Mint), Red Hat Enterprise Linux (and its derivatives such as Fedora) , and openSUSE (and its commercial derivative SUSE Linux Enterprise Server).

Linux is the most popular network operating system (NOS) runs on a server and enables the server to manage data, users, groups, security, applications, and other networking functions. It runs based on a client/server architecture in which a server enables multiple clients to share resources. Linux allows shared file and printer access

among multiple computers in a network, typically a local area network (LAN), a private network or to other networks. Linux well supports to configure and commissioning of various network servers and services such as proxy servers, Domain name systems, Mail servers, Web servers, etc that are to be accessed through internet.

Internetworking with TCP/IP

Internetworking is the practice of connecting a computer network with other networks through the use of network gateways that provide a common method of routing data packets between the networks. The most notable example of internetworking is the Internet, which a network of networks based on many underlying hardware technologies, but unified by an internetworking protocol standard, referred to as the Internet Protocol Suite and known as TCP/IP.

TCP/IP (Transmission Control Protocol (TCP) and the Internet Protocol (IP)) is a networking model and provides end-to-end connectivity specifying how data should be formatted, addressed, transmitted, routed and received at the destination. It has four abstraction layers which are used to sort all related protocols according to the scope of networking involved.

The Figure 1 shows different networks connected to internet.

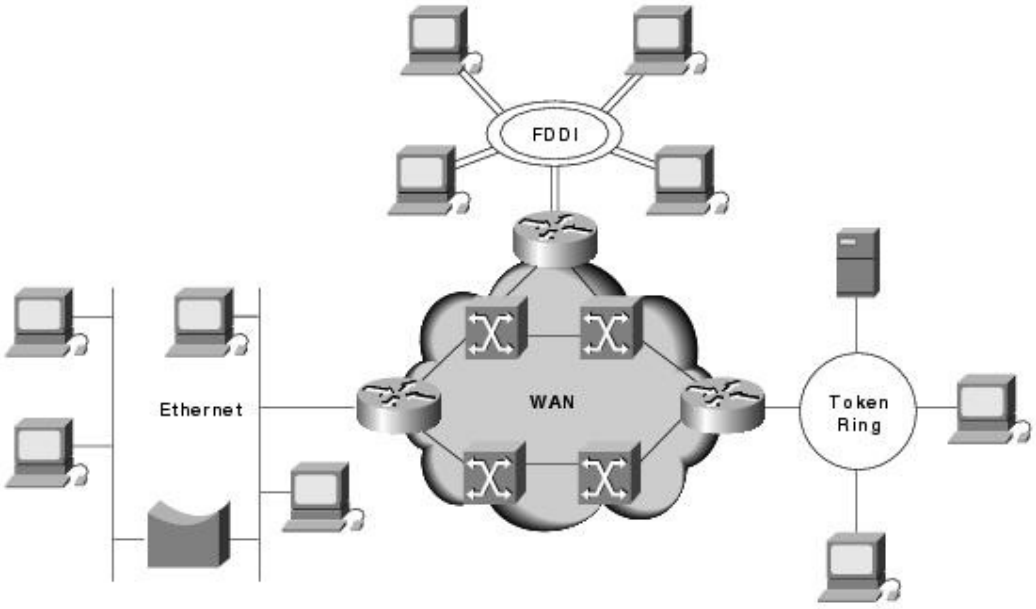


Figure 1: Different networks connected to Internet

How TCP/IP Works

TCP/IP for IP version 4 (IPv4) is a networking protocol suite that uses to communicate over the internet with other computers. It interacts with naming services like Domain Name System (DNS) and security technologies, such as IPsec for secure transfer of IP packets between computers.

Linux provides extensive support for the Transmission Control Protocol/Internet Protocol (TCP/IP) suite, as both a protocol and a set of services for connectivity and management of IP internetworks. Knowledge of the basic concepts of TCP/IP is an absolute requirement for the proper understanding of the configuration, deployment, and troubleshooting of IP-based Linux server.

TCP/IP Protocol Architecture

TCP/IP protocols map to a four-layer conceptual model. The four layers are Application, Transport, Internet, and Network Interface. Each layer corresponds to one or more layers of the seven-layer Open Systems Interconnection (OSI) model.

Figure 2 shows the TCP/IP protocol architecture.

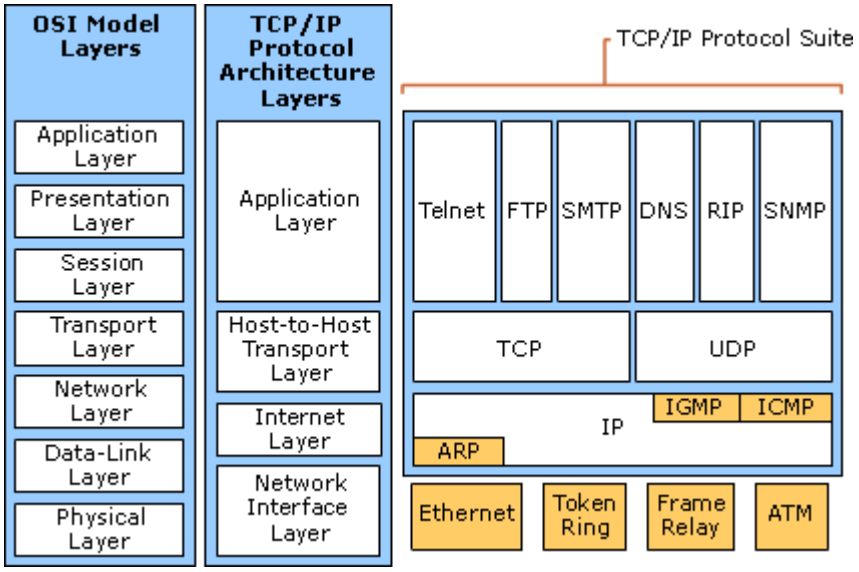


Figure 2: TCP/IP Protocol Architecture

Network Interface Layer

The Network Interface layer also called the Network Access layer that handles placing TCP/IP packets on the network medium and receiving TCP/IP packets off the network medium. TCP/IP was designed to be independent of the network access method, frame format, and medium. In this fashion, TCP/IP can be used to connect differing network types and these include local area network (LAN) media such as Ethernet and Token Ring and also WAN technologies such as X.25 and Frame Relay. The network interface layer encompasses the data link and physical layers of the OSI model.

Internet Layer

The Internet layer handles addressing, packaging, and routing functions. The core protocols of the Internet layer are IP, ARP, ICMP, and IGMP.

- The Internet Protocol (IP) is a routing protocol that handles IP addressing, routing, and the fragmentation and reassembly of packets.
- The Address Resolution Protocol (ARP) handles resolution of an Internet layer address to a Network Interface layer address, such as a hardware address.
- The Internet Control Message Protocol (ICMP) handles providing diagnostic functions and reporting errors due to the unsuccessful delivery of IP packets.
- The Internet Group Management Protocol (IGMP) handles management of IP multicast group membership.

The Internet layer is similar to the Network layer of the OSI model.

Transport Layer

The Transport layer handles and provides session and datagram communication services to Application layer. The core protocols of the Transport layer are Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

- TCP provides a one-to-one, connection-oriented, reliable communications service. TCP handles the establishment of a TCP connection, the sequencing and acknowledgment of packets sent, and the recovery of packets lost during transmission.
- UDP provides a one-to-one or one-to-many, connectionless, unreliable communications service. UDP is used when the amount of data to be transmitted is small (such as data that fits into a single packet), when you do not want the overhead of establishing a TCP connection, or when the applications or upper layer protocols provide reliable delivery.

The TCP/IP Transport layer is similar to the Transport layer of OSI model.

Application Layer

The application layer provides services for an application program to ensure that effective communication with another application program in a network is possible. The application layer is not the application itself that is doing the communication, but with various application layer protocols.

The most widely known Application layer protocols are those used for the exchange of user information:

- The Hypertext Transfer Protocol (HTTP) is used to transfer files that make up the Web pages of the World Wide Web.
- The File Transfer Protocol (FTP) is used for interactive file transfer.
- The Simple Mail Transfer Protocol (SMTP) is used for the transfer of mail messages and attachments.
- Telnet, a terminal emulation protocol, is used for logging on remotely to network hosts.

Additionally, the following Application layer protocols help facilitate the use and management of TCP/IP networks:

- The Domain Name System (DNS) is used to resolve a host name to an IP address.
- The Routing Information Protocol (RIP) is a routing protocol that routers use to exchange routing information on an IP internetwork.
- The Simple Network Management Protocol (SNMP) is used between a network management console and network devices (routers, bridges, intelligent hubs) to collect and exchange network management information.

The TCP/IP Application layer encompasses the responsibilities of the Session, Presentation, and Application layers of OSI model.

How Data Transmitted

Figure 3 shows, how data transmitted from source computer to destination computer.

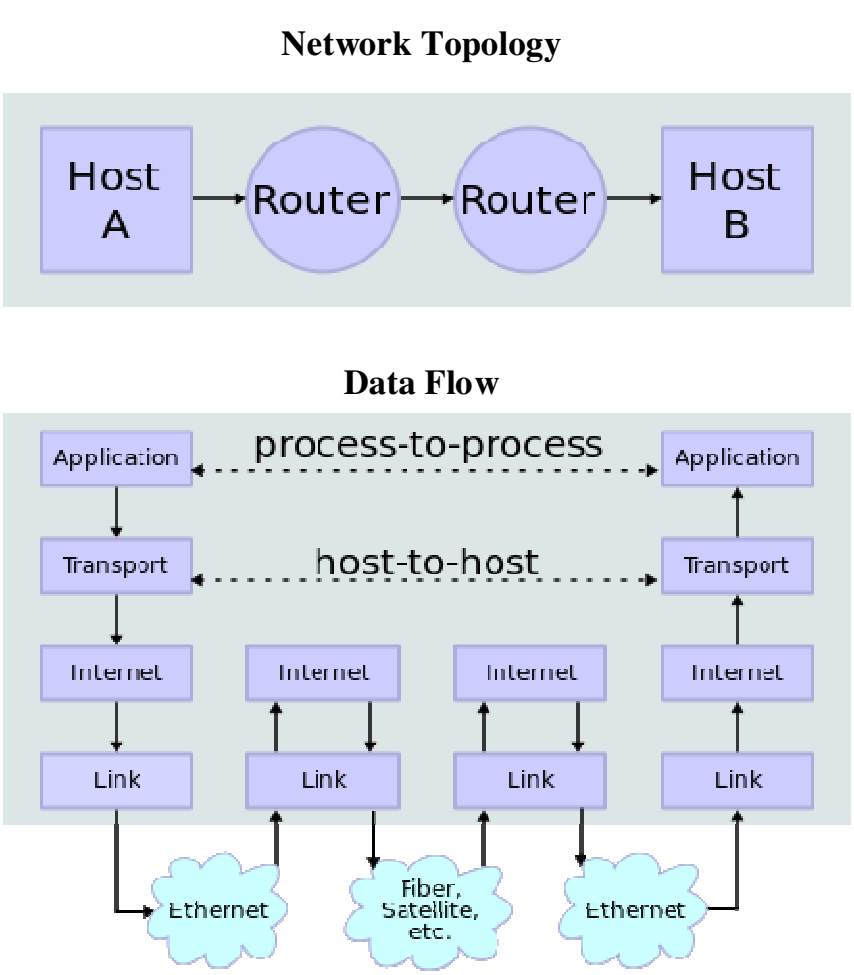


Figure 3: Data transmission between two Hosts

1.4 USING NETWORK CLIENTS

A client is a piece of computer hardware or software or both or a computer program that sends a service request to a server/computer program or accesses a service made available by a server/computer program. A network client is a client that can interact with servers/computer program through network/internet. The term client was first applied to devices that were not capable of running their own stand-alone programs, but could interact with remote computers via a network.

For example, web browsers are clients that connect to web servers and retrieve web pages for display. Email clients retrieve email from mail servers. Online chat uses a variety of clients, which vary depending on the chat protocol being used. Multiplayer video games or online video games may run as a client on each computer. The term "client" may also be applied to computers or devices that run the client software or users that use the client software. Similarly, the devices such as laptops, notebooks, palmtops, tablet PCs, smart phones, and other such devices also called network clients through which services requests can be sent or services can be retrieved to/from servers that are providing services. Figure 4 shows network clients that are being used in a cloud computing model.

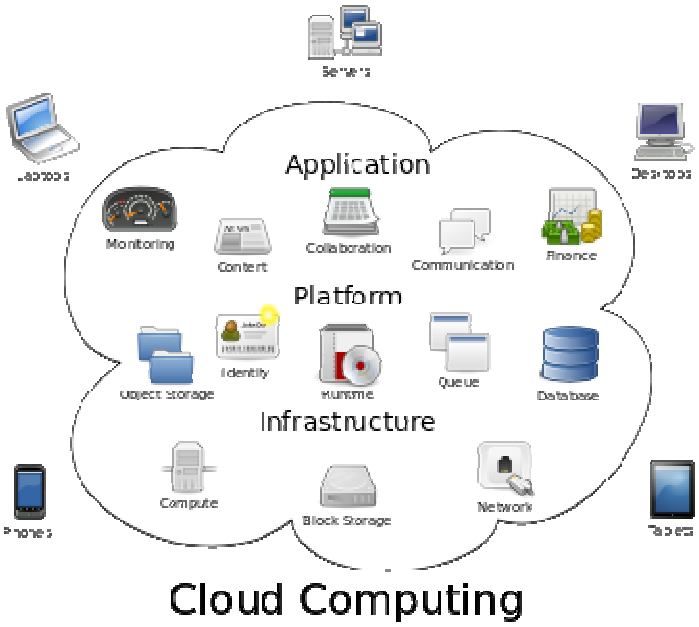


Figure 4: Network clients in a Cloud computing model

1.5 UNDERSTANDING SYSTEM INITIALIZATION

Here system means the combination of the computer hardware and the software (generally it the operating system installed on the computer hardware). Linux system (the computer installed with Linux operating system) initialization means how system gets stared after power is on. The Linux startup process is the process of Linux-operating system initialization. It is in many ways similar to the BSD and other Unix style boot processes, from which it derives.

In Linux, the flow of control during a boot is from BIOS (Basic Input/output System), to boot loader, to kernel. The kernel then starts the scheduler and runs the first program *Init* (which is mostly responsible to run startup scripts for each runlevel), at which point the kernel goes idle unless called externally.

init (short for initialization) is a program for Unix-based computer operating systems that spawns all other processes. It runs as a daemon and typically has PID 1. The *boot loader* starts the kernel and the kernel starts *init*. If some one has to delete *init* without a replacement, the system would encounter a kernel panic on the next reboot.

Overall system initialization process

i) The *BIOS* performs hardware-platform specific startup tasks

ii) Once the hardware is recognized and started correctly, the BIOS loads and executes the partition boot code from the designated boot device, which contains phase 1 of a Linux boot loader. Phase 1 loads phase 2 (the bulk of the boot loader code). Some loaders may use an intermediate phase (known as phase 1.5) to achieve this since modern large disks may not be fully readable without further code.

iii) The *boot loader* often presents the user with a menu of possible boot options. It then loads the operating system, which decompresses into memory, and sets up system functions such as essential hardware and memory paging, before calling 'start_kernel()'.

'start_kernel()' then performs the majority of system setup (interrupts, the rest of memory management, device initialization, drivers, etc.) before spawning separately, the idle process and scheduler, and the Init process (which is executed in user space).

iv) The Init process executes scripts as needed that set up all non-operating system services and structures in order to allow a user environment to be created, and then presents the user with a login screen.

The standard sequence for initializing a Linux system is as follows:

- Power on the System
- Initializing the BIOS
- Bootloader
- Kernel initialization
- Starting from "init"

Initializing the BIOS

- The BIOS (Basic Input/Output System) is the interface between the hardware and software at a very basic level, it provides all the basic instructions used by the operating system.
- The BIOS begins by executing an auto-ignition test (POST), and then it searches for devices.
- After the POST, a boot device is selected from a list that is configurable in the BIOS.
- The BIOS reads and executes the first physical sector of the boot media selected on the system, which is usually contained in the first 512 bytes of hard disk.

Bootloader

The bootloader is usually contained in the first sector of the disk and then read and executed by the BIOS. The storage space that reads the BIOS is not sufficient to contain all the bootloader, but just a part sufficient enough to start the rest of the bootloader, which is usually contained in a configuration file stored elsewhere on the disc. Hence the start is done in two steps:

- Launch via BIOS
- Launch a file under boot

The bootloader is designed to load and run the system kernel. The standard bootloader is GRUB but can also shift to LILO.

Kernel initialization

The kernel in Linux handles all operating system processes, such as memory management, task scheduling, I/O, interprocess communication, and overall system control. This is loaded in two stages - in the first stage the kernel is loaded into memory and decompressed, and a few fundamental functions such as basic memory management are set up. Control is then switched to the main kernel start process. Once the kernel is fully operational, it looks for an init process to run, which sets up a user space and the processes needed for a user environment and ultimate login. The kernel itself is then allowed to go idle, subject to calls from other processes.

The kernel initialization includes:

- The detection and initialization of devices. It means any device drivers compiled into the kernel are called and try to locate their corresponding devices.
- Mounting the root file system in read-only mode
- Loading the initial process "init"

The kernel initialization is very rapid and therefore it is very difficult to follow visually. One can read system generated log file to check what happened during kernel initialization. Generally log file can be stored under */var/log/dmesg*

Initialize "init"

Init (initialization) is the father of all processes. Its primary role is to create processes from a script stored in the file */etc/inittab*. This file usually has entries which cause *init* to spawn gettys on each line that users can log in. It also controls autonomous processes required by any particular system.

- "**init"** is the main process, it will always have a **PID** value: **1**.
- **"init"** reads its configuration from the */etc/inittab* file, that contains the settings for the system at every level of execution.

Run Levels

A run level is a software configuration of the system which allows only a selected group of processes to exist. The processes spawned by *init* for each of these run levels are defined in the /etc/inittab file.

"init" defines the following run levels:

- **Level 0:** Stop (not to be attributed to the initdefault)
- **Level 1, S:** single user mode (only the root user can log). Typically used for maintenance.
- **Level 2:** Multi-user mode without NFS network
- **Level 3:** full mode for multiple users including network
- **Level 4:** User Configurable duplicate but the level 3 by default.
- **Level 5:** X11 (including network)
- **Level 6:** Restart

Runlevel (System V)

The ability to change runlevel offers easy interaction with administrators; this allows to switch between different levels of startup.

Scripts services are in */etc/rc.d/init.d*. Each runlevel correspond to a */etc/rc.d/rcX.d* directory, where **X** is the runlevel.

System Shutdown

On shutdown, *Init* is called to close down all user space functionality in a controlled manner, again via scripted directions, following which Init terminates and the Kernel executes its own shutdown.

To stop the system, use commands like:

#Shutdown -h now

#halt

#poweroff

#init 0

Check Your Progress 1

1. Write the main responsibilities of a Network Administrator.

   ……

   ……

   ……

   ……

   ……

2. What are the various run levels of Linux system?

   ……

   ……

   ……

   ……

   ……

1.6 USER REMOTE ADMINISTRATION SERVICES AND TOOLS

Remote administration is an approach being followed to control either a computer system or a network or an application or all three from a remote location. A remote location may refer to a computer in the next room or one on the other side of the world. Generally, remote administration is essentially adopted when it is difficult or impractical to a person to be physically present and do administration on a system's terminal.

Requirements to perform Remote Administration

Network connectivity is essentially needed to perform remote administration. The network could be either Local Area Network (LAN) connectivity or internet connectivity depending on remote location. It means, any computer with an internet connection or on a LAN can be remotely administered.

For non-malicious administration, the user must install or enable remote administration software/tool on the host system then the user/client can access the host system from another computer using the remote tool.

Common Services for which remote administration is used

Generally, remote administration is needed for user management, file system management, software installation/configuration, network management- Network Security/Firewalls, VPN, Infrastructure Design, Network File Servers, Auto-mounting etc. and kernel optimization/ recompilation.

The following are some of the tasks/ services for which remote administration need to be done:

General

Controlling one's own computer from a remote location (e.g. to access the software on a personal computer from an internet café).

ICT Infrastructure Management

Remote administration essentially needed to administer the ICT infrastructure such as the servers, the routing and switching components, the security devices and other such related.

Shutdown

- Shutting down or rebooting a computer over a network

Accessing Peripherals

- Using a network device, like printer
- Retrieving streaming data, much like a CCTV system

Modifying

- Editing another computer's registry settings
- Modifying system services
- Installing software on another machine
- Modifying logical groups

Viewing

- Remotely assisting others
- Supervising computer or internet usage
- Access to a remote system's "Computer Management" snap-in

Hacking

Computers infected with malware such as Trojans sometimes open back doors into computer systems which allow malicious users to hack into and control the computer. Such users may then add, delete, modify or execute files on the computer to their own ends.

Remote Desktop Solutions for Linux

Most people who are used to a Unix-style environment know that a machine can be reached over the network at the shell level using utilities like telnet or ssh. And some people realize that X Windows output can be redirected back to the client workstation. But many people don’t realize that it is easy to use an entire desktop over the network. The following are some of proprietary and open source applications that can be used to achieve this.

SSH (Secure Shell)

Secure Shell (SSH) is a proprietary cryptographic network tool for secure data communication between two networked computers that connects, via a secure channel over an insecure network, a server and a client (running SSH server and SSH client programs, respectively). The protocol specification distinguishes between two major versions that are referred to as SSH-1 and SSH-2.

The best-known application of the tool is for access to shell accounts on Unix-like operating systems-GNU/Linux, OpenBSD, FreeBSD, but it can also be used in a similar fashion for accounts on Windows.

SSH is generally used to log into a remote machine and execute commands. It also supports tunneling, forwarding TCP ports and X11 connections, it can transfer files using the associated SSH file transfer (SFTP) or secure copy (SCP) protocols. SSH uses the client-server model.

SSH is important in cloud computing to solve connectivity problems, avoiding the security issues of exposing a cloud-based virtual machine directly on the Internet. An SSH tunnel can provide a secure path over the Internet, through a firewall to a virtual machine

OpenSSH (OpenBSD Secure Shell)

OpenSSH is a tool providing encrypted communication sessions over a computer network using the SSH protocol. It was created as an open source alternative to the proprietary Secure Shell software suite offered by SSH Communications Security.

Telnet

Telnet is used to connect a remote computer over network. It provides a bidirectional interactive text-oriented communication facility using a virtual terminal connection on internet or local area networks. Telnet provides a command-line interface on a remote host. Most network equipment and operating systems with a TCP/IP stack support a Telnet service for remote configuration (including systems based on Windows NT). Telnet is used to establish a connection to Transmission Control Protocol (TCP) on port number 23, where a Telnet server application (telnetd) is listening.

Experts in computer security, recommend that the use of Telnet for remote logins should be discontinued under all normal circumstances, for the following reasons:

- Telnet, by default, does not encrypt any data sent over the connection (including passwords), and so it is often practical to eavesdrop on the communications and use the password later for malicious purposes; anybody who has access to a router, switch, hub or gateway located on the network between the two hosts where Telnet is being used can intercept the packets passing by and obtain login, password and whatever else is typed with a packet analyzer.
- Most implementations of Telnet have no authentication that would ensure communication is carried out between the two desired hosts and not intercepted in the middle.
- Several vulnerabilities have been discovered over the years in commonly used Telnet daemons.

rlogin

rlogin is an utility for Unix-like computer operating systems that allows users to log in on another host remotely through network, communicating through TCP port 513.

rlogin has several serious security problem- all information, including passwords is transmitted in unencrypted mode. rlogin is vulnerable to interception. Due to serious security problems, rlogin was rarely used across distrusted networks (like the public internet) and even in closed networks.

rsh

The remote shell (rsh) can connect a remote host across a computer network. The remote system to which rsh connects runs the rsh daemon (rshd). The daemon typically uses the well-known Transmission Control Protocol (TCP) port number 514. In security point of view, it is not recommended.

PuTTY

PuTTY is a free and open source terminal emulator application which can act as a client for the SSH, Telnet, rlogin, and raw TCP computing protocols and as a serial console client. The name "PuTTY" has no definitive meaning, though "tty" is the name for a terminal in the Unix tradition, usually held to be short for Teletype. PuTTY was originally written for Microsoft Windows, but it has been ported to various other operating systems as well

VNC (Virtual Network Computing)

VNC is a remote display system which allows the user to view the desktop of a remote machine anywhere on the internet. It can also be directed through SSH for security

Install VNC server on a computer (server) and install client on local PC. Setup is extremely easy and server is very stable. On client side, set the resolution and connect to IP of VNC server.

One can download VNC software from the following URLs:

http://www.realvnc.com/download.html

http://www.tightvnc.com/download.html

http://www.uvnc.com/

FreeNX allows to access desktop from another computer over the internet. One can use this to login graphically to a desktop from a remote location. One example of its use would be to have a FreeNX server set up on home computer, and graphically logging in to the home computer from work computer, using a FreeNX client. One can download FreeNX software from the following URLs:

https://help.ubuntu.com/community/FreeNX

http://ubuntuforums.org/showthread.php?t=97277&highlight=freenx

http://freenx.berlios.de/ (FreeNX homepage)

Wireless Remote Administration

Remote administration software has recently started to appear on wireless devices such as the BlackBerry, Pocket PC, and Palm devices, as well as some mobile phones.

Generally these solutions do not provide the full remote access seen on software such as VNC or Terminal Services, but do allow administrators to perform a variety of tasks, such as rebooting computers, resetting passwords, and viewing system event logs, thus reducing or even eliminating the need for system administrators to carry a laptop or be within reach of the office.

AetherPal and Netop are some of the tools used for full wireless remote access and administration on Smartphone devices.

Disadvantages of Remote Administration

Remote administration has many disadvantages too apart from its advantages. The first and foremost disadvantage is the security. Generally, certain ports to be open at Server level to do remote administration. Due to open ports, the hackers/attackers takes advantage to compromise the system. It is advised that remote administration to be used only in emergency or essential situations only to do administration remotely. In normal situations, it is ideal to block the ports to avoid remote administration.

Check Your Progress 2

1. Why remote administration is needed? Explain.

   ……..

   ……..

   ……..

   ……..

   ……..

2. List the different network clients.

   ……..

   ……..

   ……..

   ……..

   ……..

3. What are the different remote administration tools?

   ……..

   ……..

   ……..

   ……..

   ……..

1.7 SUMMARY

In this unit, different roles and responsibilities of a network administrator are clearly explained The TCP/IP and its role in data transmission from source to destination is made clear. System initialization process and importance of remote administration also covered.

1.8 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1) Network Administrator is responsible to

- Ensure data network connectivity

- Network monitoring and management
- Testing the network for breaches, if any
- Keeping an eye out for needed updates
- Update Access Control Lists (ACLs) time to time to regulate network traffic
- Security controls enforcement
- Preparing and implementation of security policy and standards

2) The following are various run levels of a Linux system

Level 0: Stop (not to be attributed to the initdefault)

Level 1, S: single user mode (only the root user can log). Typically used for maintenance.

Level 2: Multi-user mode without NFS network

Level 3: full mode for multiple users including network

Level 4: User Configurable duplicate but the level 3 by default.

Level 5: X11 (including network)

Level 6: Restart

Check Your Progress 2

1) The need for remote administration is for

- User management
- ICT Infrastructure management
- Problem Diagnosis and Troubleshooting
- File system management
- Software installation/configuration
- Network Management- Network Security/Firewalls, VPN, Infrastructure Design, Network File Servers, Auto-mounting etc.

2) The different network clients are

Computers, Notebooks, Palmtops, Tablet PCs, Smart phones and other such related.

3) The following are different remote administration tools

SSH, Telnet, rsh, rlogin, openSSH, VNC, etc

1.9 FURTHER READINGS

1. Computer Networks by Andrew S Tanenbaum , Fifth Edition
2. SA2, Redhat System Administration I & II, Student Workbook
3. Cisco Certified Network Associate Study Guide, Seventh Edition by Todd Lammle
4. Redhat Enterprise Linux System Administration
5. http://en.wikipedia.org/wiki/Internetworking
6. http://en.wikipedia.org/wiki/Remote_administration

UNIT 2 NETWORK ADMINISTRATION ACTIVITIES

Structure **Page Nos.**

2.0 Introduction 20

2.1 Objectives 20

2.2 Managing Software Packages 20

2.3 File Systems 23

2.4 Managing Users 29

2.5 System and Kernel Management 34

2.6 Basic Troubleshooting 38

2.7 Summary 43

2.8 Answers to Check Your Progress 44

2.9 Further Readings 45

2.0 INTRODUCTION

System administration is a critical activity that includes installation of software , configuration, commissioning, monitoring and finally problem diagnosis & troubleshooting of systems. It is a challenging task to the system administrator to ensure the uptime of the systems all times. Creation of a suitable file system with adequate disk partitions requires the system administrator enough working knowledge. Once system commissioning is over, the administrator has to hardening of OS in terms of removing unnecessary packages that come along with OS, close unnecessary ports and other such related. System and kernel management is another activity where the administrator has to keep on doing as part o regular system resources monitoring. It is very essential to the system administrator to know and use at least some useful tools/commands that are being used as part of problem diagnosis and troubleshooting so that system health can be maintained always and in turn make the systems uptime always.

2.1 OBJECTIVES

After going through this unit, you will be able to:

- know how to manage software packages;
- understand what is a file system and its structure;
- know how to manage users;
- understand the management of system and kernel; and
- find commands that are useful for troubleshooting.

2.2 MANAGING SOFTWARE PACKAGES

Software Packages are archives of files that include all the files that make up a piece of software(such as an application itself, shared libraries, development packages containing files that needed to build software against a library, etc) and also has a set of instructions to make them work. A package is properly integrated into the

distribution it has been built for installation paths, dependencies, desktop integration, and proper startup scripts for servers, etc.

Package manager

Generally, in modern Linux distributions like openSUSE, software installation is done with a package manager. Package manager is a collection of software tools to automate the process of installing, upgrading, configuring, and removing software packages for a computer's operating system. It works on top of RPM (Red hat Package Manager) and gets software packages from repositories (such as online servers, CDs and DVDs), solves the dependencies and installs them on your system. The package manager also makes it easy to remove packages later or to update them. The number of packages available for installation depends on which repositories you have added. Package manager typically maintains a database of software dependencies and version information to prevent software mismatches and missing prerequisites.

Package management systems

Package management systems are designed to save organizations time and money through remote administration and software distribution technology that eliminate the need for manual installs and updates. This can be particularly useful for large enterprises whose operating systems are based on Linux and other Unix-like systems, typically consisting of hundreds or even thousands of distinct software packages.

Package Metadata

Packages are distributions of software, applications and data. Packages also contain metadata that contains

- Summary (software name, etc)
- Description
- A list of files contained in the package
- The version of the software it contains as well as the release number of the package
- When, where and by whom it has been built
- What architecture it has been built for
- Checksums of the files contained in the package
- The license of the software it contains
- Which other packages it requires to work properly

After installation, metadata is stored in a local package database.

Package dependencies

The important aspect of the packages archive is the relations they contain. The packages also relates files to other packages as the packaged applications need an execution environment (like other tools, libraries, etc.) to actually run the application. Package dependencies are used to express such relations.

For example, package A needs the packages of B, C and D to be installed in order to work properly.

Package dependencies are transitive, which means that when package A needs package B, and package B needs package C, package A also needs package C, which is why you sometimes end up with lots of packages to install although you just wanted that one application.

Dependencies on libraries (typically packages with a name that starts with "lib") are very common and pretty much every single application depends on a set of library packages.

Packages and package dependencies are very important aspects of Linux distributions (as well as other BSD and UNIX systems) because they provide a modular way to set up and manage an operating system and its applications. This is especially true for library packages. For example, the package `openssl` contains cryptographic libraries that are used by many applications and other libraries (e.g. for SSL encryption). When a new, improved version of `openssl` is available, all the applications that use it will benefit from it just by upgrading that single package to the newer version. It is also a very efficient way to maintain a stable and secure system. It means, when a security hole, exploit or bug affects a library used by one or many applications, upgrading the single package will fix it for all of them. Figure 1 Shows the linkages between user systems, package manager, repository, meta data, packages and package dependencies.

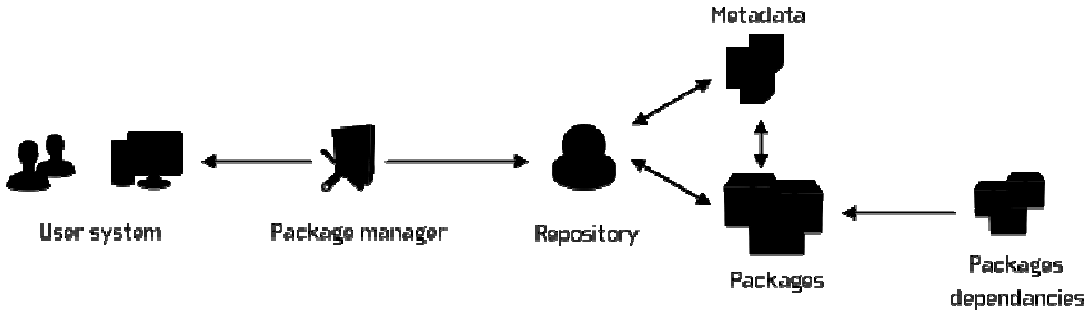


Figure 1: Linkage between user systems, package manager and repository

Package formats

In Linux distribution, the native software comes in three kinds of package formats.

- tgz (tar gzip files), which are basically source code archives and can hold anything the package maintainer thinks useful. Apart from the archive format itself, it necessary to extract the files, but there is nothing standardized about the content of a tgz file. They need to be compiled in order to run the software.

- rpm (RPM Package Manager), which are pre-compiled archives that are created by Red Hat Linux and standardized by the LSB (Linux Standard Base- to lower the overall costs of supporting the Linux platform)..

- deb (Debian) which are pre-compiled archives that are used on Debian based system.

However, if the archives format noticed the system of the required dependencies, they don't provide the dependency management capability and they will just present any encountered problem to the user at first sight, and leave it for user to decide what to do.

For instance if you want to install a RPM package **A** that has dependencies to RPM package **B**, RPM will not automatically install package **B** but just tell you that it needs package **B** and stop. It's up to the user to install package **B** and then afterwards package **A**. Now imagine package **B** has dependencies on package **C** and package **D** and package **D** has dependencies to package **E** and so on and so on. You end up chasing package dependencies manually down all the branches of this very huge tree.

RPM : rpm Package Manager

rpm is a powerful package manager, which can be used to build, install, query, verify, update, and erase individual software packages. A package consists of an archive of files and meta-data used to install and erase the archive files. The meta-data includes helper scripts, file attributes, and descriptive information about the package. Packages come in two varieties: binary packages, used to encapsulate software to be installed, and source packages, containing the source code and recipe necessary to produce binary packages.

One of the following basic modes must be selected: Query, Verify, Signature Check, Install/Upgrade/Freshen, Uninstall, Initialize Database, Rebuild Database, Resign, Add Signature, Set Owners/Groups, Show Querytags, and Show Configuration.

Use of rpm with some options

The general form of an rpm install command is

rpm {**-i|--install**} [**install-options**] *PACKAGE_FILE ...*

The general form of an rpm upgrade command is

rpm {**-U|--upgrade**} [**install-options**] *PACKAGE_FILE ...*

This upgrades or installs the package currently installed to a newer version. This is the same as install, except all other version(s) of the package are removed after the new package is installed.

rpm {**-F|--freshen**} [**install-options**] *PACKAGE_FILE ...*

This will upgrade packages, but only if an earlier version currently exists. The *PACKAGE_FILE* may be specified as an **ftp** or **http** URL, in which case the package will be downloaded before being installed.

2.3 FILE SYSTEM

A file system is a way in which files are stored, accessed, overwritten, and deleted on a media format by using a computer and the operating system (OS) installed onto that computer. Different types of OS typically have different file systems, though they are often somewhat similar in design and how files are accessed through the OS. Generally, a number of different types of file systems that have been designed and implemented, with new types being created and used by newer OS versions. A file system at its most basic level be defined as the way in which data is stored in individual files on a computer hard drive or other media, and how that data is then accessed again in the future. File system have methods and data structures that an operating system uses to keep track of files on a disk or partition and the way the files are organized on the disk. A computer user will typically work on a computer to create, alter, save, and delete a variety of files for different purposes.

Some file systems are used on local data storage devices; others provide file access via a network protocol (e.g. NFS, SMB, etc). Some file systems are "virtual", in that the "files" supplied are computed on request (e.g. procfs) or are merely a mapping into a different file system used as a backing store. The file system manages access to both the content of files and the metadata about those files. It is responsible for arranging storage space; reliability, efficiency, and tuning with regard to the physical storage medium are important design considerations.

Aspects of file systems

Following are main aspects of file systems:

Space management

File systems allocate space in a granular manner, usually multiple physical units on the device. The file system is responsible for organizing files and directories, and keep track of which areas of the media belong to which file and which are not being used.

File system fragmentation

Fragmentation occurs when unused space or single files are not contiguous. Once file system is built, files are created, modified and deleted. When a file is created, the file system allocates space for the data. Some file systems permit or require specifying an initial space allocation and subsequent incremental allocations as the file grows. If files are deleted, the space reserved for it will be allocated and use by other files. This creates used and unused areas of various sizes. This is called as a free space fragmentation. When a file is created, and there is no contiguous space available for its initial allocation, the space must be assigned in fragments. When a file is modified such that it becomes larger it may exceed the space initially fragmented.

Filenames

A filename is used to identify a storage location in the file system. Most file systems have restrictions on the length of filenames. In some file systems, filenames are case-insensitive (i.e. filenames such as ‘ABC’ and ‘abc’ refer to the same file); in others, filenames are case-sensitive (i.e. the names ‘ABC’ and ‘abc’ refer to two separate files).

Most modern file systems allow filenames to contain a wide range of characters from the Unicode character set. Most file system interface utilities have restrictions on the use of certain special characters, disallowing them within filenames (the file system may use these special characters to indicate a device, device type, directory prefix, or file type). However, these special characters might be allowed by enclosing the filename with double quotes ("). For simplicity, special characters are generally discouraged within filenames.

Directories

File systems typically have directories (also called folders) which allow the user to group files into separate collections. This may be implemented by associating the file name with an index in a table of contents or an inode in a Unix-like file system. Directory structures may be flat (i.e. linear), or allow hierarchies where directories may contain subdirectories.

Metadata

Metadata is a information that is typically associated with each file within a file system. The length of the data contained in a file may be stored as the number of blocks allocated for the file or as a byte count. The time that the file was last modified may be stored as the file's timestamp. File systems might store the file creation time, the time it was last accessed, the time the file's meta-data was changed, or the time the file was last backed up. Other information includes the file's device type (e.g. block, character, socket, subdirectory, etc.), its owner user ID and group ID, its access permissions and other file attributes (e.g. whether the file is read-only, executable, etc.).

Design limitations

All file systems have some functional limit that defines the maximum storable data capacity within that system. These functional limits are a best-guess effort by the designer based on how large the storage systems are right now and how large storage systems are likely to become in the future. Disk storage has continued to increase at near exponential rates, so after a few years, file systems have kept reaching design limitations that require computer users to repeatedly move to a newer system with ever-greater capacity.

File system complexity typically varies proportionally with the available storage capacity. The file systems of early 1980s home computers with 50 KB to 512 KB of storage would not be a reasonable choice for modern storage systems with hundreds of gigabytes of capacity. Likewise, modern file systems would not be a reasonable choice for these early systems, since the complexity of modern file system structures would consume most or all of the very limited capacity of the early storage systems.

Types of file systems

The following are some of the file system types and classified into disk/tape file systems, network file systems and special-purpose file systems.

Disk file systems

Disk file systems are the file systems which manage data on permanent storage devices. Magnetic disks are the most common of such devices. A disk file system takes advantages of the ability of disk storage media to randomly address data in a short amount of time. In disk files systems, data access is fast. It supports multiple users (or processes) to access various data on the disk. Some of the example disk file systems are FAT (FAT12, FAT16, FAT32), exFAT, NTFS, ext3, ext4, etc.

In a disk file system, there is typically a master file directory, and a map of used and free data regions. Any file additions, changes, or removals require updating the directory and the used/free maps. Random access to data regions is measured in milliseconds so this system works well for disks.

Optical file system

Optical media use different file systems than hard disks or flash media, because of the write-once nature of most optical discs. Even rewritable discs use different file systems because of the way that rewritable media is managed. ISO 9660 and Universal Disk Format (UDF) are two common formats for Compact Discs, DVDs and Blu-ray discs.

Flash file systems

A flash file system is a file system designed for storing files on flash memory devices. These are becoming more common as the number of mobile devices is increasing, the cost per memory size decreases, and the capacity of flash memories increases.

Tape file systems

A tape file system is a file system and tape format designed to store files on tape in a self-describing form. Magnetic tapes are sequential storage media with significantly longer random data access times than disks. Tape requires linear motion to wind and unwind potentially very long reels of media. This tape motion may take several seconds to several minutes to move the read/write head from one end of the tape to the other.

Network file systems

A network file system is a file system that acts as a client for a remote file access protocol, providing access to files on a server. Examples of network file systems include clients for the NFS, AFS, SMB protocols, and file-system-like clients for FTP and WebDAV(Web Distributed Authoring and Versioning). WebDAV is an extension of the Hypertext Transfer Protocol (HTTP) that facilitates collaboration between users in editing and managing documents and files stored on World Wide Web servers.

Shared disk file systems

A shared disk file system is one in which a number of machines (usually servers) have access to the same external disk subsystem (usually a SAN). The file system arbitrates access to that subsystem, preventing write collisions. Examples include GFS2 from Red Hat, GPFS from IBM, etc.

Special file systems

A special file system presents non-file elements of an operating system as files so they can be acted on using file system APIs. This is most commonly done in Unix-like operating systems, but devices are given file names in some non-Unix-like operating systems as well.

Flat file systems

A flat file system is a system of files in which every file in the system must have a different name. Flat file system stores all its files in the same directory. In this system, every file has a different name since there is only one list of files. Flat file systems cannot contain multiple tables.

In Windows 95 and most other operating system today, files are managed in a hierarchical file system with a hierarchy of directories and subdirectories, each containing a number of files (or subdirectories). The operating system allows more than one file to have the same name as long as it is stored in a different directory. Early versions of the Macintosh and DOS operating systems used a flat file system.

File systems and operating systems

Many operating systems require support for more than one file system. Sometimes the OS and the file system are so tightly interlink, due to this, it is difficult to separate out the file system.

There needs to be an interface provided by the operating system software between the user and the file system. This interface can be textual (such as provided by a command line interface, such as the Unix shell) or graphical (such as provided by a graphical user interface, such as file browsers).

Unix-like operating systems

Unix-like operating systems create a virtual file system, which makes all the files on all the devices appear to exist in a single hierarchy. This means, there is one root directory, and every file existing on the system is located under it somewhere. Unix-like systems can use a RAM disk or network shared resource as its root directory. Figure 2 shows a sample Unix directory structure.

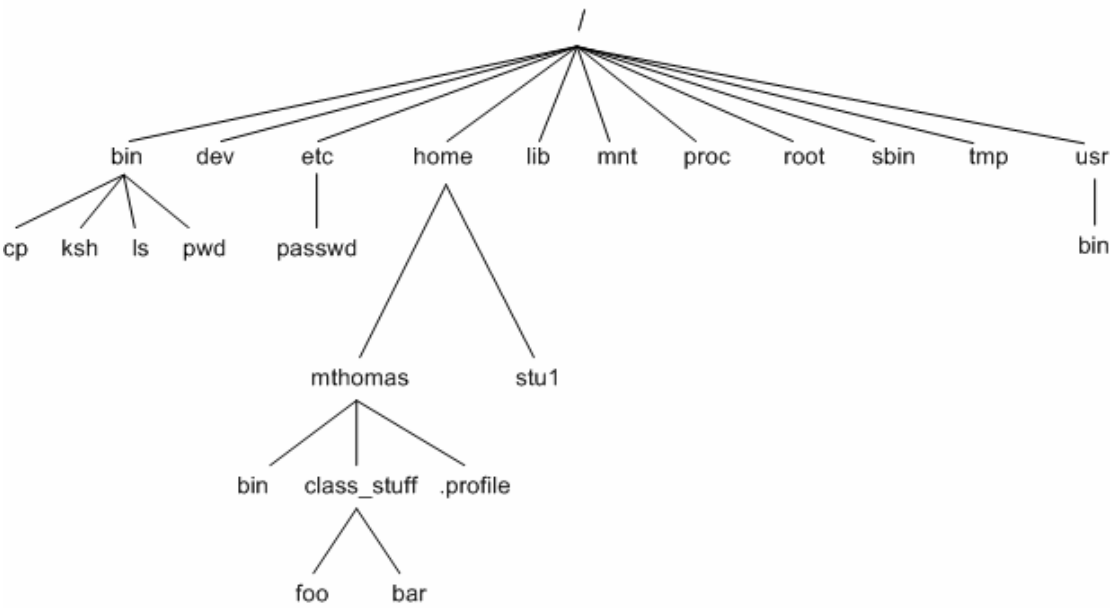


Figure 2: Unix directory Structure

In order to access a file system in UNIX, first it needs to mount it. Mounting a file system is simply making the particular file system accessible at a certain point in the Unix directory tree. When mounting a file system, it does not matter, if the file system is in a hard disk partition, CD-ROM, floppy, or USB storage device. You simply need to know the device name associated with the particular storage device and a directory you would like to mount it to. Unix-like operating systems often include software and tools that assist in the mounting process.

In many situations, file systems other than the root need to be available as soon as the operating system has booted. All Unix-like systems therefore provide a facility for mounting file systems at boot time. System administrators define these file systems in the configuration file fstab (*vfstab* in Solaris), which also indicates options and mount points.

In some situations, there is no need to mount certain file systems at boot time, although their use may be desired thereafter. There are some utilities for Unix-like systems that allow the mounting of predefined file systems upon demand.

Removable media have become very common with microcomputer platforms. They allow programs and data to be transferred between machines without a physical connection. Common examples include USB flash drives, CD-ROMs, and DVDs. Utilities have therefore been developed to detect the presence and availability of a medium and then mount that medium without any user intervention.

An automounter is a program or software facility which automatically mounts file systems in response to access operations by user programs. This is usually used for file systems on network servers, rather than relying on events such as the insertion of media, as would be appropriate for removable media.

Linux Operating System

The Linux file system is a hierarchically structured tree where every location has its distinct meaning. Linux supports many different file systems. It is similar to Unix file system. The file system is a tree-shaped structure. The root of the tree, which is called the 'file system root' but is always depicted as being above all other and is

identified by the slash character "/". It is the highest place you can go to and beneath it are almost always only directories. Figure 3 shows the sample directory of a Linux file system.

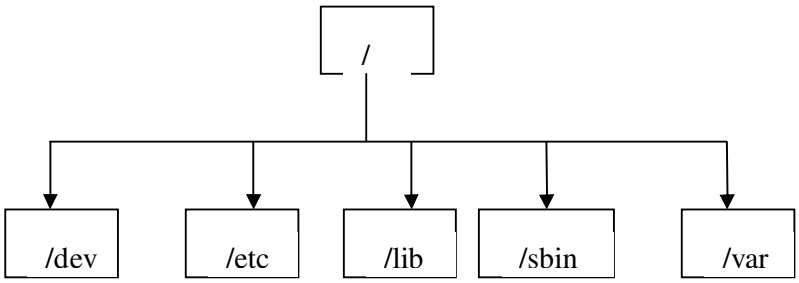


Figure 3: Sample Linux File Structure

The /dev directory contains file system entries which represent devices that are attached to the system. These files are essential for the system to function properly.

The /etc directory is reserved for configuration files that are local to your machine. No binaries are to be put in /etc.

The /lib directory should contain only those libraries that are needed to execute the binaries in /bin and /sbin.

The /proc directory contains special files that either extract information or send information to the kernel.

The /sbin directory is for executables used only by the root user.

The /var directory writes log files

Microsoft Windows File System

Windows makes use of the FAT, NTFS, exFAT and ReFS file systems. Windows uses a *drive letter* abstraction at the user level to distinguish one disk or partition from another. For example, the path `C:\WINDOWS` represents a directory `WINDOWS` on the partition represented by the letter C. Drive C is most commonly used for the primary hard disk partition , on which Windows is usually installed and from which it boots. Generally, Drive A and Drive B are used for two floppy drives. This tradition has become so firmly ingrained that bugs exist in many applications which make assumptions that the drive that the operating system is installed on is C.

File system commands

The following are some of the file system commands in Linux

- mount (to mount a file system)

syntax: #mount [-lhV]

-h prints a help message;

-V prints a version string; and just

-l lists all mounted file systems (of type type).

Note: use 'man' utility in Linux to know more on mount command

- umount (to unmount a file system)

syntax : #umount [-hV]

-V Print version and exit.

-h Print help message and exit.

Note: use ‘man’ utility in Linux to know more on umount command

- df [–hk] (to know file system disk space)

-h print sizes in human readable format (e.g., 1K 234M 2G)

-k prints in kilobytes

Note: use ‘man’ utility in Linux to know more on df command

- du [-bB] (estimate file space usage)

-b in bytes

-B in Block-size

Note: use ‘man’ utility in Linux to know more on du command

2.4 MANAGING USERS

The most important responsibilities of any system administrator is managing and monitoring of users. This section explains on how a new user account is created, deleted and other such related in a Linux operating system. The following explains on different types if users.

Users

Generally, users on a system are identified by a username and a userid. The username is something that users would normally refer to, but as far as the operating system is concerned this is referred to using the userid (or uid). The username is typically a user friendly string, such as your name, whereas the userid is a number. The userid numbers should be unique (one number per user).

Special Users

There are some default users on all systems when first installed. Other than the root user however these can normally be disabled from logging in. One should be more careful about deleting usernames as sometimes these are used by different tasks running on the system. Special users normally have names such as sys, bin, adm etc.

The root user has an userid of 0, which has a special meaning. The root user has full permissions to do anything on the system. It is not bound by any of the permissions on the system. There are some tasks that can only be performed by root. To use root permissions, one would normally login under a normal userid and “su” to root or use sudo as required.

Switch User (su)

One of the features of Linux is the ability to change userid when logged into a system. The command ‘su’ is sometimes referred to as superuser, however this is not completely correct. In the early days of UNIX it was only possible to change to the root user, which made for the superuser command however it is now possible to change to any user using the su command. It is more correct to refer to the command as the switch user command.

The switch user command "su" is used to change between different users on a system, without having to logout. The most common use is to change to the root user, but it can be used to switch to any user depending upon the user’s settings. To switch to a different user other than root, then the username is used as the last option on the command.

With regard to user accounts there are three basic types of Linux user accounts: administrative (root), regular, and service.

The Linux administrative root account is automatically created at the time of installation of Linux, and it has administrative privileges for all services on Linux Operating System. The root account is also known as super user. Regular users have the necessary privileges to perform standard tasks on a Linux computer such as running word processors, databases, and Web browsers. Regular users can store files in their own home directories. Since regular users do not normally have administrative privileges, it is not possible to delete critical operating system configuration files. Services such as Apache, Squid, mail, games, and printing have their own individual service accounts. These accounts exist to allow each of these services to interact with your computer.

Each user on a Red Hat Enterprise Linux system is assigned a unique user identification number, also known as a UID. UIDs below 500 are reserved for system users such as the root user and service users.

User group

A user group is a group of one or more users. A user can be a member of more than one group. In Red Hat Enterprise Linux, when a user is added, a private user group (primary group) is created—meaning that a user group of the same name is created and that the new user is the sole user in that group.

The user information is stored in the system files such as /etc/passwd and /etc/shadow files, and that additionally, group membership information is stored in the /etc/group file. The following explains on how user accounts created, modified and deleted with appropriate commands along with examples:

Understand fields in /etc/passwd

The password file name in Linux is ‘passwd’ and is stored in /etc directory. The path of a password file in Linux is ‘/etc/passwd’.

The /etc/passwd file stores essential information, which is required during login i.e. user account information. The structure of each entry (record) in a password file is in a seven(7) colon format. The /etc/passwd contains one entry per line for each user (or user account) of the system. All fields are separated by a colon (:) symbol. Total seven fields as follows.

Here is an example of a ***passwd*** file:

root:x:0:0:root:/root:/bin/bash

bin:x:1:1:bin:/bin:/bin/bash

daemon:x:2:2:daemon:/sbin:/bin/bash

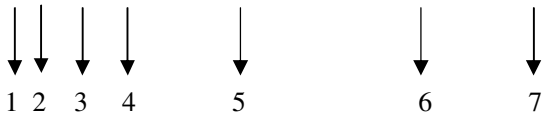
news:x:9:13:News system:/etc/news:/bin/bash

uucp:x:10:14::/var/lib/uucp/taylor_config:/bin/bash

cquoi:x:500:100:Cool.....:/home/cquoi:/bin/bash

Generally, passwd file entry looks like the following:

murli:x:1025:1024:A Murli M Rao:/usr1/murli:/bin/bash



1) Username: It is used when user logs in. It should be between 1 and 32 characters in length. (ex. murli)
2) Password: An x character indicates that encrypted password is stored in /etc/shadow file.
3) User ID (UID)**:** Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups. (ex. 1025)
4) Group ID (GID): The primary group ID (stored in /etc/group file) (ex. 1024)
5) User ID Info**:** The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. (ex. A Murli M Rao)
6) Home directory: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /(ex. /usr1/murli)
7) Command/shell: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. (ex. /bin/bash)

Check Your Progress 1

1. What are the common metadata in packages?

……

……

……

……

……

……

2. Explain the different parts of a passwd file.

……

……

……

……

……

……

User Administration

Add New User in Linux

Command: useradd (Enables a super user or root to create a new user or updates default new user information)

Syntax

Useradd [options] <user-name>

The following is more in details

useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g initial_group] [-G group[,...]] [-m [-k skeleton_dir]] [-p passwd] [-s shell] [-u uid [-o]] login

useradd -D [-g default_group] [-b default_home] [-f default_inactive] [-e default_expire_date] [-s default_shell]

Example

#useradd murli (creates a user murli with default shell, hole directory, UID, GID automatically)

(or) one can use various options as listed above to have specific values

User Account Modification

Command: usermod (Enables a super user or root user to modify a users account)

Syntax

Usermod [options]< user-name>

The following is more in details

usermod [-c comment] [-d home_dir [-m]] [-e expire_date] [-f inactive_time] [-g initial_group] [-G group[,...]] [-l login_name] [-p passwd] [-s shell] [-u uid [-o]] [-L|-U] login

#usermod -G others admin murli

Delete User account in Linux

Command: userdel (Enables a super user to remove a users account)

Syntax

userdel [-r] login

[–r Files in the user's home directory will be removed along with the home directory itself and the user's mail spool. Files located in other file systems will have to be searched for and deleted manually]

Syntax

Example

#userdel –r murli (the user account 'murli' can be deleted)

Groups Administration in Linux

The file **/etc/group** contains a list of the users who belong to the different groups. As a matter of fact, whenever a large number of users may have access to the system, they are frequently placed in different groups, each of which has it own access rights to the files and directories.

It has different fields that are separated by "**:**":

groupe_name : special_field : group_number : member1, member2

Here is an example of a ***/etc/group*** file:

root:x:0:root

bin:x:1:root,bin,daemon

daemon:x:2:

tty:x:5:

disk:x:6:

lp:x:7:

wwwadmin:x:8:

kmem:x:9:

wheel:x:10:

mail:x:12:cyrus

news:x:13:news

Group Creation

A system administrator can manage a group's account. The various tasks that a system administrator can perform include adding, modifying and deleting group account.

Command: groupadd (Creates a new group account)

Syntax

groupadd [-g gid [-o]] group

-g gid	The numerical value of the group's ID. This value must be unique, unless the -o option (override) is used . The value must be non-negative. The default is to use the smallest ID value greater than 99 and greater than every other group. Values between 0 and 99 are typically reserved for system accounts.

Example

#groupadd test1

This will create a test1 group

Group Modification

Command: groupmod (Enables a super user or root to modify a group)

Syntax

groupmod [-g gid [-o]] [-n group_name] group

-g gid	The numerical value of the group's ID. This value must be unique, unless the -o option (override) is used. The value must be non-negative. Values between 0 and 99 are typically reserved for system groups. Any files which the old group ID is the file group ID must have the file group ID changed manually.
-n group_name	The name of the group will be changed from group to group_name.

Example

#groupmod test1 test2 (This will modify group name test1 to test2)

Group Deletion

Command: groupdel (The name of the group you wish to delete.)

Syntax

groupdel groupname

Example

#groupdel test2 (this will delete the test2 group)

2.5 SYSTEM AND KERNEL MANAGEMENT

An operating system is the first piece of software that the computer executes when you turn the machine on. The operating system loads itself into memory and begins managing the resources available on the computer. It then provides those resources to other applications that the user wants to execute.

System Management

Once system is booted, the operating system has to perform various tasks such as task scheduling, memory management, disk management , I/O and security management.

The following explains on typical services that an operating system provides.

Task scheduler

The task scheduler is able to allocate the execution of the CPU to a number of different tasks. Some of those tasks are the different applications that the user is running, and some of them are operating system tasks. The task scheduler is the part of the operating system that lets you print a document from your word processor in one window while you are downloading a file in another window and recalculating a spreadsheet in a third window.

Memory manager

The memory manager controls the system's RAM and normally creates a larger virtual memory space using a file on the hard disk.

Disk manager

The disk manager creates and maintains the directories and files on the disk. When you request a file, the disk manager brings it in from the disk.

Network manager

The network manager controls all data moving between the computer and the network.

Other I/O services manager

The OS manages the keyboard, mouse, video display, printers, etc.

Security manager

The OS maintains the security of the information in the computer's files and controls who can access the computer.

System Management Commands

The following are some of the system management commands:

- who (who is logged in)

Syntax: who [-abd]

-a, --all

-b, --boot

time of last system boot

-d, --dead

print dead processes

Example: [murli@imssit ~]$ who

murli pts/0 Jun 2 14:15 (10.10.115.3)

murli pts/1 Jun 2 15:44 (10.10.115.3)

Note: use ‘man’ utility in Linux to know more on mount command

- pwd (displays path of current working directory)

Syntax: pwd

Example: [murli@imssit ~]$ pwd

/home/murli

Note: use ‘man’ utility in Linux to know more on mount command

Other such related command ,which are to be used as part of system management are the following:

- top, vmstat, ps, kill, df, du, mount, unount, tar, cpio, head, tail, mkdir, chdir, chown, chgrp, chmod, nice, find, grep, etc.

Note: use ‘man’ utility in Linux to know more on al he above commands

Kernel Management

The kernel is the main component of most computer operating systems. It is a bridge between applications and the actual data processing done at the hardware level. The kernel's responsibilities include managing the system's resources (the communication between hardware and software components). Usually, as a basic component of an operating system, a kernel can provide the lowest-level abstraction layer for the resources (especially processors and I/O devices) that application software must control to perform its function. It typically makes these facilities available to application processes through inter-process communication mechanisms and system calls. Figure 4 shows the kernel and its interactive components.

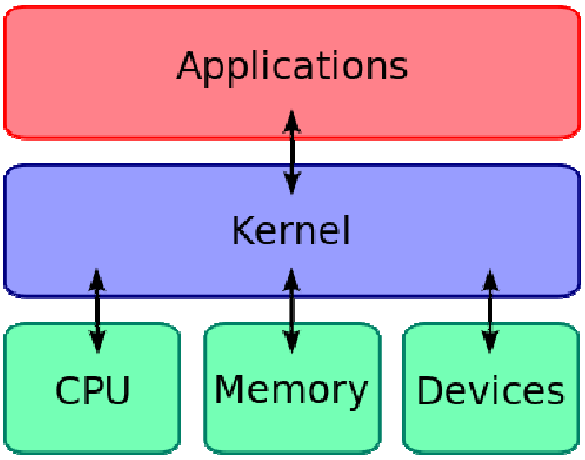


Figure 4: Kernel and its interactive components

Operating system tasks are done differently by different kernels(monolithic and micro kernels), depending on their design and implementation. While monolithic kernels execute all the operating system code in the same address space to increase the performance of the system, micro kernels run most of the operating system services in user space as servers, aiming to improve maintainability and modularity of the operating system. A range of possibilities exists between these two extremes.

The kernel's primary function is to manage the computer's hardware and resources and allow other programs to run and use these resources. Typically, the resources consist of:

The Central Processing Unit

This is the most central part of a computer system, responsible for running or executing programs. The kernel takes responsibility for deciding at any time which of the many running programs should be allocated to the processor or processors (each of which can usually run only one program at a time)

The computer's memory

Memory is used to store both program instructions and data. Typically, both need to be present in memory in order for a program to execute. Often multiple programs will want access to memory, frequently demanding more memory than the computer has available. The kernel is responsible for deciding which memory each process can use, and determining what to do when not enough is available.

Input/Output (I/O)

The I/O devices present in the computer, such as keyboard, mouse, disk drives, USB devices, printers, displays, network adapters, etc. The kernel allocates requests from applications to perform I/O to an appropriate device and provides convenient methods for using the device.

Inter-process communication (IPC)

Kernels usually provide methods for synchronization and communication between processes called inter-process communication (IPC).A kernel may implement these features itself, or rely on some of the processes it runs to provide the facilities to other processes, although in this case it must provide some means of IPC to allow processes to access the facilities provided by each other. Finally, a kernel must provide running programs with a method to make requests to access these facilities.

Linux Kernel Management

The Linux kernel allows drivers and features to be compiled as modules rather than as part of the kernel itself. This means that users can often change features in the kernel or add drivers without recompiling, and that the Linux kernel doesn't have to carry a lot of unnecessary baggage.

The following explains on how to see the already loaded in running kernel, how to add modules to the kernel and finally on how to remove modules from kernel.

What is Loaded?

The first utilities that you'll want to get to know are lsmod and modinfo. Open a terminal and run lsmod. Note that you won't need to use sudo or log in as root just to probe the modules in the system.

You'll see output like this when you use lsmod:

```
Module          Size  Used by
parport_pc      18855  0
ppdev            5030  0
lp               7462  0
sco              7209  2
parport         27954  3 parport_pc,ppdev,lp
bridge          39630  0
stp              1440  1 bridge
bnep             9427  2
```

This shows the modules that are loaded, their size, and whether they're being used by other modules. Take the parport module, for instance. It's being used by several other modules, but what are they? The modinfo utility will tell us. To do so run the following:

Run modinfo parport, and you'll see something like this:

```
filename:       /lib/modules/2.6.32-5-amd64/kernel/drivers/parport/parport.ko
license:        GPL
depends:
vermagic:       2.6.32-5-amd64 SMP mod_unload modversions
```

It tells us where the module is found, and its license, and that it has no dependencies.

Removing Modules

Modules can be removed using the rmmod utility. The usage is simple, just rmmod *modulename*. However, if we try to remove the parport module, we get this error:

```
ERROR: Module parport is in use by parport_pc,ppdev,lp
```

You can force module removal using rmmod -f, but that's not a good idea, usually. A better way to do it is to use modprobe -r which will automatically look to see what other modules depend on it, and unload those modules as well. If they're in use, then modprobe will refuse to remove them as well, unless you use the -f option with modprobe too.

Installing Modules

What if you have a module you want to load into the kernel? You can do that with insmod or modprobe.

The preferred method is modprobe, because it will also load any modules that the requested module depends on. For instance, if I didn't have the parport module loaded and went to load the lp or parport_pc modules, modprobe would go ahead and load parport as well.

To load a module using modprobe run modprobe *modulename*.

Blacklisting Modules

You may on occasion need to "blacklist" a module. Why would you need this feature? Sometimes a module will cause a conflict with another module, is superseded by another module, or is otherwise undesirable.

To blacklist a module, the easiest way to do it (there's usually more than one way to do things...) is to add the module to /etc/modprobe.d/blacklist.conf. For instance, on Debian systems the evbug module is automatically blacklisted because it's not something most users will need. To add a module to the blacklist, just add one line to the blacklist.conf file:

blacklist *modulename*

2.6 BASIC TROUBLESHOOTING

Troubleshooting is a form of problem solving, often applied to repair failed products or processes. It is a logical, systematic search for the source of a problem so that it can be solved, and so the product or process can be made operational again. Troubleshooting is needed to develop and maintain complex systems where the symptoms of a problem can have many possible causes.

The following are some of the commands being used in a Linux environment for problem diagnosis and troubleshooting:

- Use tail –f to watch log file in real time, advantage is simple you can spot error or warning message in real time.

Command : **# tail –f /var/log/maillog**

- Use telnet command to see if you get response or not. Sometime you will also see some informative message:

telnet ip port

Example(s):

 # telnet localhost 53
 # telnet localhost 25

- Make sure you can see PID of your service.

pidof service-name

cat /var/run/service.pid

Example(s):

 # pidof sshd
 # cat /var/run/sshd.pid

- You need to make sure that your DNS server or third party DNS server (ISP) is accessible. This is an important step, as many network services depend upon DNS; especially sendmail/postfix or Squid etc for example. Run dig or nslookup. No timeout should occur.

  # dig your-domain.com
  # nslookup gw.isp.com
  # more /etc/resolv.conf

- For networking troubleshooting, make sure your ip address configuration is right, gateway, routing, hostname etc all configured. Here is list of tools on RedHat Linux to verify or modify information:

Hostname verification or setup tools

- **hostname** : To get hostname of server.
- **more /etc/sysconfig/network** : To see hostname and networking details
- **more /etc/hosts** :Make sure at least localhost entry do exist.

Ethernet configuration tools

- **ifconfig** : To see running network card information.
- **ifconfig eth0 up|down** : To enable|disable network interface
- **service network reload|restart|stop|start** : To reload (after changed made in ip config file)|restart|stop|start network interface with all properties.
- **route|netstat –rn** : To print routing table
- **ping ip-address** : To see if host is alive or dead
- **more /etc/modules.conf** : To see your network card configuration alias for eth0 exists or not.
- **lsmod** : To list loaded modules (read as drivers), here you need to see that eth0 module is loaded or not, if not loaded then use insmod to insert (load) driver.
- **dhclient** : Dynamic Host Configuration Protocol Client, run this if your Ethernet card is not getting ip from DHCP box on startup; this command does by default shows useful information.

To see if service blocked because of access control

- **iptables –n –L** : To list all iptable rules; useful to see if firewall blocks service or not.
- **service iptables stop|start** : To start|stop iptables
- **more /etc/xinetd.conf**

OR

- **more /etc/xinetd.conf/SERVICENAME** = To list configuration of xinetd server. Again useful to see if firewall xinetd based security blocks service or not (xinetd includes host-based and time-based access control)
- **more /etc/hosts.allow** : To see list of hosts allowed to access service.
- **more /etc/hosts.deny** : To see list of hosts NOT allowed to access service. **NOTE** first TCP wrappers (hosts.allow|hosts.deny) checked and then xinetd-based access control checked.
- **more /etc/path/to/application.conf** : See your application configuration file for access control. For example smb.conf and many other applications/services got own access control list in application. You need to check that as well.

Some more commands

Getting ram information

#cat /proc/meminfo

or if you want to get just the amount of ram you can do:

#cat /proc/meminfo | head -n 1

Getting cpu info

Sometimes in troubleshooting we want to know what processor we are dealing with along with how much cpu is currently being used by our OS and programs. We can do this with these two commands.

#cat /proc/cpuinfo

#top

Example :[murli@imssit ~]$ top

Output

top - 14:25:45 up 10 days, 21:07, 1 user, load average: 0.00, 0.05, 0.02

Mem: 4147692k total, 4124540k used, 23152k free, 57604k buffers

Swap: 10241396k total, 562856k used, 9678540k free, 2250180k cached

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
5511	oracle	16	0	1280m	264m	262m	S	0.3	6.5	4:08.74	oracle
5665	oracle	16	0	1280m	314m	311m	S	0.3	7.8	12:13.93	oracle
17154	murli	16	0	2776	1048	772	R	0.3	0.0	0:00.16	top
1	root	16	0	2900	540	464	S	0.0	0.0	0:01.38	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.31	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	RT	0	0	0	0	S	0.0	0.0	0:00.20	migration/1
5	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/1
6	root	RT	0	0	0	0	S	0.0	0.0	0:00.22	migration/2
7	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/2
8	root	RT	0	0	0	0	S	0.0	0.0	0:00.12	migration/3
9	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/3

```
 10 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 events/0
 11 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 events/1
 12 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 events/2
 13 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 events/3
 14 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 khelper
 15 root      5 -10     0    0    0 S  0.0  0.0   0:00.00 kthread
```

Check out how much hard drive space is left

#df -h

Example :[murli@imssit ~]$ df –h

Output

```
Filesystem            Size  Used Avail Use% Mounted on
/dev/cciss/c0d0p5      16G  2.3G   13G  16% /
/dev/cciss/c0d0p1     289M   28M  247M  10% /boot
none                  2.0G     0  2.0G   0% /dev/shm
/dev/cciss/c0d0p7     7.7G   51M  7.3G   1% /tmp
/dev/cciss/c0d0p2     202G  182G   11G  95% /u01
/dev/cciss/c0d0p3      29G   21G  6.6G  76% /u02
/dev/cciss/c0d0p8     4.9G  2.8G  1.9G  60% /usr
```

Check out how much disk partitions usage

#df –k

Example: [murli@imssit ~]$ df -k

Output

```
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/cciss/c0d0p5     16126420   2307900  12999208  16% /
/dev/cciss/c0d0p1       295561     27863    252438  10% /boot
none                   2073844         0   2073844   0% /dev/shm
/dev/cciss/c0d0p7      8064272     51480   7603140   1% /tmp
/dev/cciss/c0d0p2    211663320 190318352  10593068  95% /u01
/dev/cciss/c0d0p3     30233928  21790480   6907636  76% /u02
/dev/cciss/c0d0p8      5036284   2854352   1926100  60% /usr
```

To estimate file space usame

#du

Example: [murli@imssit ~]$ du

Output

```
16      ./.kde/Autostart
24      ./.kde
16      ./.xemacs
104     .
```

Kill a process

#ps -A | grep ProgramName

#kill 7207

Show all network connections

#netstat

Example: [murli@imssit ~]$ netstat

Output

Active Internet connections (w/o servers)

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State
tcp	0	0	imssit:64734	imssit:1521	ESTABLISHED
tcp	0	0	imssit:64742	imssit:1521	ESTABLISHED
tcp	0	0	imssit:1521	imssit:32786	ESTABLISHED
tcp	0	0	imssit:1521	imssit:40238	ESTABLISHED

List all files that are currently open on the system

#ls –of

Example: [murli@imssit ~]$ ls –of

Output

.. . .bash_logout .bashrc .emacs .zshrc
.bash_history .kde .gtkrc .xemacs .bash_profile

List all processes running

#ps-aef

Example: [murli@imssit ~]$ ps –aef

Output

UID	PID	PPID	C	STIME	TTY	TIME	CMD
root	1	0	0	May22	?	00:00:01	init [5]
root	2	1	0	May22	?	00:00:00	[migration/0]
root	3	1	0	May22	?	00:00:00	[ksoftirqd/0]
root	4	1	0	May22	?	00:00:00	[migration/1]
root	5	1	0	May22	?	00:00:00	[ksoftirqd/1]
root	6	1	0	May22	?	00:00:00	[migration/2]
root	7	1	0	May22	?	00:00:00	[ksoftirqd/2]
root	8	1	0	May22	?	00:00:00	[migration/3]
root	9	1	0	May22	?	00:00:00	[ksoftirqd/3]
root	10	1	0	May22	?	00:00:00	[events/0]
root	11	1	0	May22	?	00:00:00	[events/1]
root	12	1	0	May22	?	00:00:00	[events/2]
root	13	1	0	May22	?	00:00:00	[events/3]
root	14	1	0	May22	?	00:00:00	[khelper]

Check Your Progress 2

1. Write the syntax and uses of "useradd" command in Linux.

   ……..

   ……..

   ……..

   ……..

   ……..

2. Which command is used to display real-time running tasks in a Linux environment? Explain the significance of this command using an example.

   ……..

   ……..

   ……..

   ……..

   ……..

3. How the “Disc Uses” is checked in Linux? Explain with an example.

   ……..

   ……..

   ……..

   ……..

   ……..

2.7 SUMMARY

In this unit, the need of a package manager for installation and un-install a package is clearly explained. Different types of file systems and their need and usage was discussed. The unit also covered on how to manage user accounts and also on system and kernel management. Finally the unit explained with some examples on troubleshooting commands.

2.8 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1. Packages contains metadata that contains
   - Summary (software name, etc)
   - Description
   - A list of files contained in the package
   - The version of the software it contains as well as the release number of the package
   - When, where and by whom it has been built

- What architecture it has been built for
- Checksums of the files contained in the package
- The license of the software it contains
- Which other packages it requires to work properly

2. Generally, passwd file entry looks like the following:

murli:x:1025:1024:A Murli M Rao:/usr1/murli:/bin/bash

1. Username: It is used when user logs in. It should be between 1 and 32 characters in length. (ex. murli)
2. Password: An x character indicates that encrypted password is stored in /etc/shadow file.
3. User ID (UID)**:** Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups. (ex. 1025)
4. Group ID (GID): The primary group ID (stored in /etc/group file) (ex. 1024)
5. User ID Info**:** The comment field. It allow you to add extra information about the users such as user's full name, phone number etc. (ex. A Murli M Rao)
6. Home directory: The absolute path to the directory the user will be in when they log in. If this directory does not exists then users directory becomes /(ex. /usr1/murli)
7. Command/shell: The absolute path of a command or shell (/bin/bash). Typically, this is a shell. (ex. /bin/bash)

Check Your Progress 2

1. 'useradd' is a command to create a new user in Linux environment

Syntax is

Useradd [options] <user-name>

or

useradd [-c comment] [-d home_dir] [-e expire_date] [-f inactive_time] [-g initial_group] [-G group[,...]] [-m [-k skeleton_dir]] [-p passwd] [-s shell] [-u uid [-o]] login

Example:# useradd murli

2. 'top' is a command to display real-time running tasks in a Linux environment

Example :[murli@imssit ~]$ top

Output

top - 14:25:45 up 10 days, 21:07, 1 user, load average: 0.00, 0.05, 0.02

Mem: 4147692k total, 4124540k used, 23152k free, 57604k buffers

Swap: 10241396k total, 562856k used, 9678540k free, 2250180k cached

```
  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 5511 oracle    16   0 1280m 264m 262m S  0.3  6.5   4:08.74 oracle
 5665 oracle    16   0 1280m 314m 311m S  0.3  7.8  12:13.93 oracle
17154 murli     16   0  2776 1048  772 R  0.3  0.0   0:00.16 top
    1 root      16   0  2900  540  464 S  0.0  0.0   0:01.38 init
    2 root      RT   0     0    0    0 S  0.0  0.0   0:00.31 migration/0
    3 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/0
    4 root      RT   0     0    0    0 S  0.0  0.0   0:00.20 migration/1
    5 root      34  19     0    0    0 S  0.0  0.0   0:00.00 ksoftirqd/1
```

3. ‘df-k’ is a command to know the disk usage

Example: [murli@imssit ~]$ df -k

Output

```
Filesystem           1K-blocks      Used Available Use% Mounted on
/dev/cciss/c0d0p5     16126420   2307900  12999208  16% /
/dev/cciss/c0d0p1       295561     27863    252438  10% /boot
none                   2073844         0   2073844   0% /dev/shm
/dev/cciss/c0d0p7      8064272     51480   7603140   1% /tmp
/dev/cciss/c0d0p2    211663320 190318352  10593068  95% /u01
/dev/cciss/c0d0p3     30233928  21790480   6907636  76% /u02
/dev/cciss/c0d0p8      5036284
```

2.9 FURTHER READINGS

1. Computer Networks by Andrew S Tanenbaum , Fifth Edition
2. SA2, Redhat System Administration I & II, Student Workbook
3. Cisco Certified Network Associate Study Guide, Seventh Edition by Todd Lammle
4. Redhat Enterprise Linux System Administration
5. http://en.wikipedia.org/wiki/linux
6. http://en.wikipedia.org/wiki/kernel

UNIT 3 NETWORK CONFIGURATION AND SETTING

Structure **Page Nos.**

3.0	Introduction	46
3.1	Objectives	46
3.2	Configuring Networks	46
3.3	Dynamic Host Configuration Protocol (DHCP)	47
3.4	Network Domain System (DNS)	50
3.5	Network File System (NFS)	56
3.6	Web Server	63
3.7	Summary	67
3.8	Answers to Check Your Progress	68
3.9	Further Readings	69

3.0 INTRODUCTION

A computer network is a telecommunications network that allows computers to exchange data. The physical connection between networked computing devices is established using either cable media or wireless media. The best-known computer network is the Internet.

Network configuration is an activity to properly configure any network infrastructure through which various network applications/services can be run and accessed. Administrators must be able to configure IP addresses as well as other configuration files w.r.t different network services such as Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), Email, Web Servers and other such related network services.

In this unit, you will learn configuration settings of various network services such as Dynamic Host Protocol (DHCP), Domain Name System (DNS) Network File System (NFS) and Web Server

3.1 OBJECTIVES

After going through this unit, you will be able to:

- know how to install various network services;
- configure a Dynamic Host Control Protocol;
- understand and configure a Domain Name System; and
- know on how to configure a Samba server.

3.2 CONFIGURING NETWORKS

A computer network facilitates interpersonal communications, allows sharing of files, and allows sharing of network and computing resources and other such related. To do so, there is an essential need to configure various network services those facilitate for the above network applications.

Network configuration and setup of various services in any organization is a challenging task to configure various network services such as DHCP, DNS, Web Service, Email, etc to run various applications that are to be accessed through network. The following section explains various core and essential network services that are to be required in any organization through which various applications can be run and accessed through an organizational network.

3.3 DYNAMIC HOST CONTROL PROTOCOL

The Dynamic Host Configuration Protocol (DHCP) is a network protocol used to configure devices that are connected to a network. These devices can communicate on that network using the Internet Protocol (IP). It involves clients and a server operating in a client-server model.

Dynamic Host Configuration Protocol (DHCP) automatically assigns IP addresses and other network configuration information (subnet mask, broadcast address, etc) to computers on a network. The DHCP server maintains a database of available IP addresses and configuration information. A client configured for DHCP will send out a broadcast request to the DHCP server requesting an address. The DHCP server will then issue a "lease" and assign it to that client. The time period of a valid lease can be specified on the server. DHCP reduces the amount of time required to configure clients and allows one to move a computer to various networks and be configured with the appropriate IP address, gateway and subnet mask.

How DHCP Works?

The following are the activities between DHCP Server and DHCP Client.

- Lease Request: Client broadcasts request to DHCP server with a source address of 0.0.0.0 and a destination address of 255.255.255.255. The request includes the MAC address which is used to direct the reply.
- IP lease offer: DHCP server replies with an IP address, subnet mask, network gateway, name of the domain, name servers, duration of the lease and the IP address of the DHCP server.
- Lease Selection: Client receives offer and broadcasts to al DHCP servers that will accept given offer so that other DHCP server need not make an offer.
- The DHCP server then sends an acknowledgement to the client. The client is configured to use TCP/IP.
- Lease Renewal: When half of the lease time has expired, the client will issue a new request to the DHCP server.

Download and Install the DHCP Package

Most RedHat and Fedora Linux software product packages are available in the RPM format, whereas Debian and Ubuntu Linux use DEB format installation files. When searching for these packages, remember that the filename usually starts with the software package name and is followed by a version number, for example the file name as ‘dhcp-3.23.58-4.i386.rpm’.

Managing the DHCP daemon is easy to do, but the procedure differs between Linux distributions. Here are some things to keep in mind.

- Firstly, different Linux distributions use different daemon management systems. Each system has its own set of commands to do similar operations.
- Secondly, the daemon name needs to be known and in this case the name of the daemon is dhcpd.

dhcpd.conf File

You can define your server configuration parameters in the dhcpd.conf file which may be located in the /etc the /etc/dhcpd or /etc/dhcp3 directories depending on your version of Linux.

The dhcpd.conf configuration file formats in Debian / Ubuntu and Redhat / Fedora are identical.

Here is a quick explanation of the dhcpd.conf file.

How do you configure DHCP?

The following is the dhcpd.conf file

```
ddns-update-style interim

ignore client-updates


subnet 192.168.1.0 netmask 255.255.255.0 {

  # The range of IP addresses the server
  # will issue to DHCP enabled PC clients
  # booting up on the network

  range 192.168.1.201 192.168.1.220;

  # Set the amount of time in seconds that
  # a client may keep the IP address

 default-lease-time 86400;
 max-lease-time 86400;

  # Set the default gateway to be used by
  # the PC clients

  option routers 192.168.1.1;
  # Don't forward DHCP requests from this
  # NIC interface to any other NIC
  # interfaces

  option ip-forwarding off;

  # Set the broadcast address and subnet mask
  # to be used by the DHCP clients

 option broadcast-address 192.168.1.255;
 option subnet-mask 255.255.255.0;

  # Set the NTP server to be used by the
  # DHCP clients

 option ntp-servers 192.168.1.100;

  # Set the DNS server to be used by the
  # DHCP clients

 option domain-name-servers 192.168.1.100;

  # If you specify a WINS server for your Windows clients,
```

```
  # you need to include the following option in the dhcpd.conf file:
 option netbios-name-servers 192.168.1.100;

  # You can also assign specific IP addresses based on the clients'
  # ethernet MAC address as follows (Host's name is "laser-printer":

 host laser-printer {
    hardware ethernet 08:00:2b:4c:59:23;
   fixed-address 192.168.1.222;
  }
}
#
# List an unused interface here
#
subnet 192.168.2.0 netmask 255.255.255.0 {
}
```

DHCP Servers with Multiple NICs

DHCP servers with multiple interfaces pose two configuration challenges. The first is setting up the correct routing and the second is making sure only the required interfaces are listening to serve DHCP.

Routing

When a DHCP configured PC boots, it requests its IP address from the DHCP server. It does this by sending a standardized DHCP broadcast request packet to the DHCP server with a source IP address of 255.255.255.255.

If your DHCP server has more than one interface, you have to add a route for this 255.255.255.255 address so that it knows the interface on which to send the reply; if not, it sends it to the default gateway.

You can temporarily add a route to 255.255.255.255 using the following route add command:

```
root# route add -host 255.255.255.255 dev eth0
```

Create a permanent route to 255.255.255.255. This will vary according to your version of Linux

In Fedora / RedHat ,add the route to your /etc/sysconfig/network-scripts/route-eth0 file, if the route needs to be added to your eth0 interface. The following is an example:

```
#
# vi  /etc/sysconfig/network-scripts/route-eth0 and add the following entry and then save
#

255.255.255.255/32 dev eth0
```

in Ubuntu / Debian, add the route to your /etc/network/interfaces file. In this case the route is added to the eth0 interface. The following is an example:

```
#
# vi/etc/network/interfaces and add the following and then save
#
iface eth0 inet static
      up route add -host 255.255.255.255 eth0
```

Listening

Once you have defined the interface for your DHCP routing, you should also ensure that your DHCP server only listens on that interface and no others. This methodology to do this varies depending on your version of Linux.

In Fedora / RedHat, the /etc/sysconfig/dhcpd file must be edited and the DHCPDARGS variable edited to include the preferred interface. Fox example interface eth0 is preferred.

vi /etc/sysconfig/dhcpd then add the following and save

DHCPDARGS=eth1

In Debian / Ubuntu the /etc/default/dhcp3-server file must be edited and the INTERFACES variable edited to include the preferred interface. For example interface eth0 is preferred.

vi /etc/default/dhcp3-server then add the following and save

INTERFACES="eth0"

You will be able to verify success in one of two ways. First the netstat command using the –au options will give the list of interfaces listening on the bootp (DHCP) UDP port. The following is an example:

root# netstat -au | grep bootp

udp 0 0 192.168.1.100:bootps *:*

root#

Secondly, your /var/log/messages file will also reveal the following defined interfaces used when the DHCPd daemon was restarted.

Jun 8 17:22:44 dhcpd: Listening on LPF/eth0/00:e0:18:5c:d8:41/192.168.1.0/24

Jun 8 17:22:44 dhcpd: Sending on LPF/eth0/00:e0:18:5c:d8:41/192.168.1.0/24

Once, the above messages revealed when DHCPd daemon was started, the configuration is success then go for launch.

Note: Client systems running on Linux/Windows shall be configured to use DHCP.

3.4 DOMAIN NAME SYSTEM (DNS)

A DNS server, or name server, is used to resolve an IP address to a hostname or vice versa.

It is a hierarchical distributed naming system for computers, services, or any resource connected to the internet or a private network. It associates various information with domain names assigned to each of the participating entities. Most prominently, it translates easily memorized domain names to the numerical IP addresses needed for the purpose of locating computer services and devices worldwide.

By setting up a DNS server, you become part of a hierarchy of DNS servers that make up the internet. At the top of this hierarchy is the root server, represented by a dot (“.”) below the root server are the top level domains (such as .com,org,and so on).

Understanding DNS

The basic function of name server is to answer queries by providing the information that those queries request. A DNS name server primarily translates domain and host names into IP addresses. Each domain is typically represented by a least two DNS servers. The following are different types of DNS servers:

- **Primary (master) name server** contains authoritative information about the domains that it serves. In response to queries for information about its domains, this server provides that information marked as being authoritative. The primary is the ultimate source for data about the domain. The secondary name server only carries the same authority in that it has received and loaded a complete set of domain information from the primary.

- **Secondary (slave) name server** gets all information for the domain from the primary. As is the case for the primary, DNS considers the secondary information about the domain that it serves authoritative.

- **Caching name server** simply caches the information it receives about the locations of hosts and domains. It holds information that it obtains from other authoritative servers and reuses that information until the information expires.

- **Forwarding name server** is essentially a caching name server but is useful in cases where computers lie behind a firewall and in which only one computer can make DNS queries outside that firewall on behalf of all the internal computers.

Understanding Bind

Red Hat Linux (and most other Linux and UNIX systems) implement DNS services by using the Berkeley Internet Name Domain (BIND) software. The Internet software Consortium maintains BIND (at www.isc.org/products/BIND). The basic components of BIND include the following:

- **DNS server daemon** (/usr/sbin/named): the named daemon listens on a port for DNS services requests and then fulfills those requests based on information in the configuration files that you create. Mostly named receives requests to resolve the host names in your domain to IP address.

- **DNS configuration files** (named.conf and /var/named/ *): the/etc/named.conf file is where you add most of the general configuration information that you need to define the DNS services for your domain. Separate files in the /var/named directory contain specific zone information.

- **DNS lookup tools** to check that your DNS server is resolving host names properly. These include commands such as host, dig, and nslookup (which are part of the bind-utils software package).

To maintain your DNS server correctly, you can also perform the following configuration tasks with your DNS server:

Logging indicates what you want to log and where log files reside.)

Remote server options can set options for specific DNS servers to perform such tasks as blocking information from a bad server, setting encryption keys to you use with a server, or defining transfer methods)

There is no need to give out DNS information to everyone who requests it. Restrict access to those who request it based on the following.

Access control list can contain those hosts, domains or IP addresses that one wants to group together and apply the same level of access to DNS server. C acl records to group those addresses, and then indicate what domain information the locations in that acl can or can't access.

Listen-on ports by default, name server accepts only name server requests that come to port 53 on name server. You can add more port numbers if you want your name server to accept name-service queries on different ports.

Authentication is to verify the identities of hosts that are requesting services from DNS server, can use keys for authentication and authorization. (the key and trusted-keys statements are used for authentication.)

Quick-starting A DNS Server

The DNS server software that comes with the current Red Had Linux is Berkeley Internet name Daemon (BIND). The following components are required to configure BIND.

- **Configuration file** (/etc/named.conf) is the main DNS server configuration file.
- **Zone directory** (/var/named) is the directory containing files that keep information about the zone that you create for your DNS server.
- **Daemon process** (/usr/sbin/named) is the daemon process that listens for DNS requests and responds with information that the named.conf file presents.
- **Debugging tools** (named –checkconf, and named-checkzone) are to determine whetherthe created DNS configuration correctly.

The following are the points one has to keep it in mind in creating a DNS server:

- identifying your DNS servers
- creating DNS configuration files (named.conf and /var/names/*)
- starting the named daemon
- Monitoring named activities

Configure DNS Server

The following is a step by step procedure to configure a master DNS server.

For this activity, use three systems- one Linux server, second Linux clients and third window clients.

Step 1 - bind and caching-nameserver rpm is required to configure DNS. Check them for install, if not found then install

```
[root@Server ~]# rpm -qa bind*
bind-libs-9.3.3-10.el5
bind-chroot-9.3.3-10.el5
bind-devel-9.3.3-10.el5
bind-utils-9.3.3-10.el5
bind-libbind-devel-9.3.3-10.el5
bind-9.3.3-10.el5
bind-sdb-9.3.3-10.el5
[root@Server ~]# rpm -qa cach*
caching-nameserver-9.3.3-10.el5
cachefilesd-0.8-2.el5
[root@Server ~]# _
```

Step 2 - set hostname to server.example.com and ip address to 192.168.0.254

```
[root@Server ~]# cat /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=Server.example.com

[root@Server ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:11:AD:E1
          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:
          inet6 addr: fe80::20c:29ff:fe11:ade1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:99 errors:0 dropped:0 overruns:0 carrier
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:17981 (17.5 KiB)
          Interrupt:67 Base address:0x2000
```

Main configuration file for dns server is named.conf. By default, this file is not created in /var/named/chroot/etc/ directory. Instead of named.conf , a sample file /var/named/chroot/etc/named.caching-nameserver.conf is created. This file is used to make a caching only name server. You can also do editing in this file after changing its name to named.conf to configure master dns server or you can manually create a new named.conf file.

In our example, create a new named.conf file

```
[root@Server etc]# vi /var/named/chroot/etc/named.conf _
```

Or do editing exactly as shown here in image

```
options{
        directory  "/var/named/";
};

zone "example.com" {
       type master;
       file "example.com.zone";
       allow-transfer {192.168.0.1;};
};
zone "0.168.192.in-addr.arpa" {
      type master;
      file "0.168.192.in-addr.arpa.zone";
};
```

save this file with :wq and exit

Configure zone file

Defined two zone files example.com.zone for forward zone and 0.168.192.in-addr.arpa for reverse zone. These files will be stored in /var/named/chroot/var/named/ location.

Use two sample files for creating these files.

```
[root@Server named]# cd /var/named/chroot/var/named
[root@Server named]# cp localhost.zone example.com.zone
[root@Server named]# cp named.local 0.168.192.in-addr.arpa.zone
[root@Server named]# _
[root@Server named]# vi example.com.zone _
```

By default this file will look like this

```
$TTL    86400
@               IN SOA  @       root (
                                        42              ; serial
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

                IN NS           @
                IN A            127.0.0.1
                IN AAAA         ::1
```

Change this file exactly as shown in image below

```
$TTL    86400
@               SOA         example.com.      root (
                                        42              ; serial
                                        3H              ; refresh
                                        15M             ; retry
                                        1W              ; expiry
                                        1D )            ; minimum

@               NS              server.example.com.
@               NS              client1.client.com.
server          A               192.168.0.254
client1         A               192.168.0.1
client2         A               192.168.0.2
```

Now open reverse lookup zone file 0.168.192.in-addr.arpa

By default it will look like this

```
$TTL    86400
@               SOA     example.com. root.server.example.com.  (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum

          IN      NS      server.example.com
254       IN      PTR     server.example.com.
1         IN      PTR     client1.example.com.
2         IN      PTR     client2.
```

Change this file exactly as shown in image below

```
$TTL    86400
@               SOA     example.com. root.server.example.com.  (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum

          IN      NS      server.example.com
254       IN      PTR     server.example.com.
1         IN      PTR     client1.example.com.
2         IN      PTR     client2.
```

Now changed the ownership of these zone files to named group

```
[root@Server named]# chgrp named example.com.zone
[root@Server named]# chgrp named 0.168.192.in-addr.arpa.zone
[root@Server named]# _
```

Now start the named service

```
[root@Server named]# chkconfig named on
[root@Server named]# service named restart
Stopping named:                                            [  OK  ]
Starting named:                                            [  OK  ]
[root@Server named]# _
```

If service restart without any error, it means successfully configured master name server.

Check Your Progress 1

1. Discuss the activities between DHCP Server and DHCP Client.

……

……

……

……

2. What are the different types of DNS servers? Explain.

……

……

……

……

……

3.5 NETWORK FILE SYSTEM

Network File System (NFS) is a server-client protocol for sharing files between computers on a common network. It allows a computer to access directories on remote computers by mounting them on a local file system as if they were a local disk The administrator on the NFS server has to define the directories that need to be activated, or exported, for access by the NFS clients, and administrators on the clients need to define both the NFS server and the subset of its exported directories to use. It is available on a variety of UNIX based operating systems, not just Linux. The server and client do not have to use the same operating system. The client system just needs to be ruining an NFS client compatible with the NFS server.

General rules to be followed

One should follow some general rules when configuring NFS.

- Only export directories beneath the / directory.
- Do not export a subdirectory of a directory that has already been exported. The exception being when the subdirectory is on a different physical device. Likewise, do not export the parent of a subdirectory unless it is on a separate device.
- Only export local filesystems.

  When mounting any filesystem on a directory, one should know that the original contents of the directory are ignored, or obscured, in favor of the files in the mounted filesystem. When the filesystem is unmounted, then the original files in the directory reappear unchanged.

Some NFS key concepts

The following are some key NFS background concepts that will help in overall understanding.

The Virtual File System (VFS)

VFS interface is the mechanism used by NFS to redirect all access to NFS-mounted files to the remote server. It is done in such a way that files on the remote NFS server appear to the user like those on a local disk.

Stateless Operation

Programs that read and write to files on a local filesystem rely on the operating system to track their access location within the file with a pointer. As NFS is a network-based file system, and networks can be unreliable, it was decided that the NFS client daemon would act as a failsafe intermediary between regular programs running on the NFS client and the NFS server.

Normally, when a server fails, file accesses timeout and the file pointers are reset to zero. With NFS, the NFS server doesn't maintain the file pointer information, the NFS client does. This means that if an NFS server suddenly fails, the NFS client can precisely restart the file access once more after patiently waiting until the server returns online.

Caching

NFS clients typically request more data than they need and cache the results in memory locally so that further sequential access of the data can be done locally instead of access from server. This is also known as a read ahead cache. Caching therefore helps to reduce the amount of network traffic while simultaneously improving the speed of some types of data access. The NFS server caches information too, such as the directory information for the most recently accessed files and a read ahead cache for recently read files.

NFS and Symbolic links

One has to be careful with the use of symbolic links on exported NFS directories. If an absolute link points to a directory on the NFS server that hasn't been exported, then the NFS client won't be able to access it. Mounting a filesystem on a symbolic link actually mounts the filesystem on the target of the symbolic link. Plan carefully before doing this.

NFS Background mounting

NFS clients use remote procedure call (RPC) suite of network application helper programs to mount remote filesystems. If the mount cannot occur during the default RPC timeout period, then the client retries the mount process until the NFS number of retires has been exceeded. The default is 10,000 minutes, which is approximately a week. The difficulty here is that if the NFS server is unavailable, the mount command will hang for a week until it returns online.

Hard and Soft mounts

The process of continuous retrying, whether in the background or foreground, is called a hard mount. NFS attempts to guarantee the consistency of data with these constant retries. With soft mounts, repeated RPC failures cause the NFS operation to fail not hang and data consistency is therefore not guaranteed. The advantage is that the operation completes quickly, whether it fails or not. In such case it is better not to place critical data that needs to be updated regularly or executable programs in such a location. NFS has currently various versions such as version 2,3 and 4. Version 1 was a prototype.

NFS Daemons

NFS isn't a single program, but a suite of interrelated programs that work together to get the job done. The following are several daemons that are started when a system goes into run level 3 or multi-user mode. The mountd and nfsd daemons are run on systems that are servers. The automatic startup of the server daemons depends on the existence of entries that are labeled with the NFS file-system type in /etc/dfs/sharetab. To support NFS file locking, the lockd and statd daemons are run on NFS clients and servers.

i) automountd Daemon

This daemon handles the mounting and unmounting requests from the autofs service. The syntax of the command is as follows:

automountd [**-Tnv**] [-D *name*=*value*]

The command behaves in the following ways:

–T enables tracing.

-n disables browsing on all autofs nodes.

-v selects to log all status messages to the console.

-D *name*=*value* substitutes *value* for the automount map variable that is indicated by *name*.

The default value for the automount map is /etc/auto_master. Use the **-T** option for troubleshooting

ii) *lockd Daemon*

This daemon supports record-locking operations on NFS files. The lockd daemon manages RPC connections between the client and the server for the Network Lock Manager (NLM) protocol.

iii) *mountd Daemon*

This daemon handles file-system mount requests from remote systems and provides access control. The mountd daemon checks /etc/dfs/sharetab to determine which file systems are available for remote mounting and which systems are allowed to do the remote mounting. One can use the **-v** option and the **-r** option with this command.

The **-v** option runs the command in verbose mode. Every time an NFS server determines the access that a client should be granted, a message is printed on the console. The information that is generated can be useful when trying to determine why a client cannot access a file system.

The **-r** option rejects all future mount requests from clients. This option does not affect clients that already have a file system mounted.

iv) nfs4cbd Daemon

This daemon is for the exclusive use of the NFS version 4 client that manages the communication endpoints for the NFS version 4 callback program. The daemon has no user-accessible interface.

v) nfslogd Daemon

This daemon provides operational logging. NFS operations that are logged against a server are based on the configuration options that are defined in /etc/default/nfslogd. When NFS server logging is enabled, records of all RPC operations on a selected file system are written to a buffer file by the kernel

vi) statd Daemon

This daemon works with lockd daemon to provide crash and recovery functions for the lock manager. The statd daemon tracks the clients that hold locks on an NFS server. If a server crashes, on rebooting, statd daemon on the server contacts statd on the client. The client statd can then attempt to reclaim any locks on the server. The client statd also informs the server statd when a client has crashed so that the client's locks on the server can be cleared.

vii) rpcbind: (portmap in older versions of Linux)

The primary daemon upon which all the others rely, rpcbind manages connections for applications that use the RPC specification. By default, rpcbind listens to TCP port 111 on which an initial connection is made. This is then used to negotiate a range of TCP ports, usually above port 1024, to be used for subsequent data transfers. You need to run rpcbind on both the NFS server and client.

viii) nfs

Starts the RPC processes needed to serve shared NFS file systems. The nfs daemon needs to be run on the NFS server only.

ix) nfslock

Used to allow NFS clients to lock files on the server via RPC processes. The nfslock daemon needs to be run on both the NFS server and client.

x) netfs

It allows RPC processes run on NFS clients to mount NFS filesystems on the server. The netfs daemon needs to be run on the NFS client only.

Installing NFS

RedHat Linux installs nfs by default, and nfs is also activated when the system boots. One can check whether nfs installed or not using the RPM command in conjunction with the grep command to search for all installed nfs packages.

The following is an example to check:

[root# rpm -qa | grep nfs

redhat-config-nfs-1.1.3-1

nfs-utils-1.0.1-3.9

[root#

A blank list means that there is a need to install the required packages.

There is also a need to have the RPC rpcbind package installed, and the rpm command can tell whether it is installed or not. Use rpm in conjunction with grep, to check all the rpcbind applications installed or not.

The following is an example:

[root# rpm -q rpcbind

rpcbind-4.0-57

[root#

A blank list means that there is a need to install the required packages.

If nfs and rpcbind are not installed, they can be added fairly easily once find the nfs-utils and rpcbind RPMs. Remember that RPM filenames usually start with the software's name and a version number, as in nfs-utils-1.1.3-1.i386.rpm.

A Typical NFS Scenario

A small office has an old Linux server that is running out of disk space. The office cannot tolerate any down time, even after hours, because the server is accessed by overseas programmers and clients at nights and local ones by day. Budgets are tight and the company needs a quick solution until it can get a purchase order approved for a hardware upgrade. Another Linux server on the network has additional disk capacity (for ex. at /data partition) and the office would like to expand into it as an interim expansion NFS server.

Configuring NFS on The Server

Both the NFS server and NFS client have to have parts of the NFS package installed and running. The server needs rpcbind, nfs, and nfslock operational, as well as a correctly configured /etc/exports file.

The following explains how exports file is to be configured:

The /etc/exports File

The /etc/exports file is the main NFS configuration file, and it consists of two columns. The first column lists the directories you want to make available to the network. The second column has two parts. The first part lists the networks or DNS domains that can get access to the directory, and the second part lists NFS options in brackets.

For example for the scenario as mentioned above, suppose requires the following:

Read-only access to the /data/files directory to all networks

Read/write access to the /home directory from all servers on the xxx.xxx.1.0 /24 network, which is all addresses from xxx.xxx.1.0 to xxx.xxx.1.255

Read/write access to the /data/test directory from servers in the my-site.com DNS domain

Read/write access to the /data/database directory from a single server xxx.xxx.1.203.

In all cases, use the sync option to ensure that file data cached in memory is automatically written to the disk after the completion of any disk data copying operation.

```
#/etc/exports
/data/files         *(ro,sync)
/home               xxx.xxx.1.0/24(rw,sync)
/data/test          *.my-site.com(rw,sync)
/data/database      xxx.xxx.1.203/32(rw,sync)
```

After configuring /etc/exports file, there is a need to activate the settings, but first make sure that NFS is running correctly.

Starting NFS on the Server

Configuring an NFS server is straightforward:

Use the chkconfig command to configure the required nfs and RPC rpcbind daemons to start at boot and also activate NFS file locking to reduce the risk of corrupted data.

The following is the example:

```
root@# chkconfig  nfs on
root@# chkconfig  nfslock on
root@# chkconfig  rpcbind on
```

Use the init scripts in the /etc/init.d directory to start the nfs and RPC rpcbind daemons.

The following are examples to use the start option, but by using stop and restart options one can stop or restart the service as when needed.

```
root# service rpcbind start
root# service nfs start
root# service nfslock start


root# service rpcbind stop
root# service rpcbind restart
```

Test whether NFS is running correctly with the rpcinfo command or not. The following I an example to list the running RPC programs.

```
root# rpcinfo -p localhost
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100021    1   udp   1024  nlockmgr
    100021    3   udp   1024  nlockmgr
    100021    4   udp   1024  nlockmgr
    100005    1   udp   1042  mountd
    100005    1   tcp   2342  mountd
    100005    2   udp   1042  mountd
    100005    2   tcp   2342  mountd
    100005    3   udp   1042  mountd
    100005    3   tcp   2342  mountd
root#
```

Accessing NFS Server Directories from the Client

In most cases, users want their NFS directories to be permanently mounted. This requires an entry in the /etc/fstab file in addition to the creation of the mount point directory.

The /etc/fstab File

The /etc/fstab file lists all the partitions that need to be auto-mounted when the system boots. Edit the /etc/fstab file for NFS directory to be made permanently available to users on the NFS.

For example, mount the /data/files directory on server test(IP address xxx.xxx.1.100) as NFS-type filesystem using the local /mnt/nfs mount point directory.

```
#/etc/fstab
#Directory                Mount Point    Type   Options       Dump   FSCK
xxx.xxx.1.100:/data/files   /mnt/nfs       nfs    soft,nfsvers=2  0      0
```

This example is used the soft and nfsvers options; The table 1 outlines these and other useful NFS mounting options:

Table 1: NFS Option

Option	Description
bg	Retry mounting in the background if mounting initially fails
fg	Mount in the foreground
soft	Use soft mounting
hard	Use hard mounting
rsize=n	The amount of data NFS will attempt to access per read operation. The default is dependent on the kernel. For NFS version 2, set it to 8192 to assure maximum throughput.
wsize=n	The amount of data NFS will attempt to access per write operation. The default is dependent on the kernel. For NFS version 2, set it to 8192 to assure maximum throughput.
nfsvers=n	The version of NFS the mount command should attempt to use
tcp	Attempt to mount the filesystem using TCP packets: the default is UDP.
intr	If the filesystem is hard mounted and the mount times out, allow for the process to be aborted using the usual methods such as CTRL-C and the kill command.

Configuring NFS on the Client

NFS configuration on the client requires to start the NFS application; create a directory on which to mount the NFS server's directories that are exported via the /etc/exports file, and finally to mount the NFS server's directory on local system's directory, or mount point.

Starting NFS on the Client

Use the following steps to configure NFS on the client.

Use the chkconfig command to configure the required nfs and RPC rpcbind daemons to start at boot. Activate nfslock to lock the files and reduce the risk of corrupted data.

root# chkconfig netfs on

root# chkconfig nfslock on

root# chkconfig rpcbind on

Use the init scripts in the /etc/init.d directory to start the nfs and RPC rpcbind daemons. Use the following example les to start, stop and restart the processes:

root# service rpcbind start

root# service netfs start

```
root# service nfslock start
root# service rpcbind stop
root# service rpcbind restart
```

Test NFS whether is running correctly or not with the rpcinfo command. The flowing is an examplw:

.

```
root# rpcinfo -p
   program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  32768  status
    100024    1   tcp  32768  status
    100021    1   udp  32769  nlockmgr
    100021    3   udp  32769  nlockmgr
    100021    4   udp  32769  nlockmgr
    100021    1   tcp  32769  nlockmgr
    100021    3   tcp  32769  nlockmgr
    100021    4   tcp  32769  nlockmgr
    391002    2   tcp  32770  sgi_fam
root#
```

3.6 WEB SERVER

The primary function of a web server is to cater web page to the request of clients using the Hypertext Transfer Protocol (HTTP). This means delivery of HTML documents and any additional content that may be included by a document, such as images, style sheets and scripts. The server that sends your web browser the code to display a web page is called a web server. There are countless web servers all over the Internet serving countless websites to people all over the world. Whether you need a web server to host a website on the Internet a Red Hat Enterprise Linux server can function as a web server using the Apache HTTP server. The Apache HTTP server is a popular, open source server application that runs on many UNIX-based systems as well as Microsoft Windows.

A user agent, commonly a web browser initiates communication by making a request for a specific resource using HTTP and the server responds with the content of that resource or displays an error message, if not available. The resource is typically a real file on the server's secondary storage, but this is not necessarily the case and depends on how the web server is implemented.

Samba Server

Samba is a software package that comes with Red Hat Linux to share file systems and printers on a network with computers that use the session massage block (SMS) protocol. SMB is the protocol that is delivered with windows operating systems for sharing files and printers. In Red Hat Linux, the Samba software package contains a variety of daemon processes, administrative tools, user tools, and configuration files.

The default Samba configuration file is smb.conf, which is in /etc/samba directory (/etc/samba/smb.conf). If you need to access features that are not available through the samba server configuration file you can edit /etc/samba/smb.conf file as required.

Daemon processes consist of smbd (the SMB daemon) and nmbd (the NetBIOS name server). The smbd is what makes the file sharing and printing services you add to your Red Hat Linux computer available to windows client computers. The following are some of the clients that Samba supports:

Window 9X

Window 2000

Window NT

Window ME

Window XP

Window for workgroups

Ms Client 3.0 for DOS

OS/2

Dave for Macintosh computer

Samba for Linux

As for administrative tools for samba, you have several shell commands at your disposal. You can check your configuration file using the testparm and testprns commands. The smbstatus command tells you which computers are currently connected to your shared

Samba configuration file (smb.conf)

```
# smb.conf is the main Samba configuration file. You find a full commented
# version at /usr/share/doc/packages/samba/examples/smb.conf.SUSE if the
# samba-doc package is installed.
# Date: 2008-05-28
[global]
   workgroup = WORKGROUP
   printing = cups
   printcap name = cups
   printcap cache time = 750
   cups options = raw
   map to guest = Bad User
   include = /etc/samba/dhcp.conf
   logon path = \\%L\profiles\.msprofile
   logon home = \\%L\%U\.9xprofile
   logon drive = P:
   usershare allow guests = Yes
   add machine script = /usr/sbin/useradd  -c Machine -d /var/lib/nobody -s /bin/false
%m$
   domain logons = No
   domain master = No
   security = user
   usershare max shares = 100
[homes]
   comment = Home Directories
   valid users = %S, %D%w%S
   browseable = No
   read only = No
   inherit acls = Yes
```

```
[profiles]
   comment = Network Profiles Service
   path = %H
   read only = No
   store dos attributes = Yes
   create mask = 0600
   directory mask = 0700
[users]
   comment = All users
   path = /home
   read only = No
   inherit acls = Yes
   veto files = /aquota.user/groups/shares/
[groups]
   comment = All groups
   path = /home/groups
   read only = No
   inherit acls = Yes
[printers]
   comment = All Printers
   path = /var/tmp
   printable = Yes
   create mask = 0600
   browseable = No
[print$]
   comment = Printer Drivers
   path = /var/lib/samba/drivers
   write list = @ntadmin root
   force group = ntadmin
   create mask = 0664
   directory mask = 0775

## Share disabled by YaST
# [netlogon]

[public]
   comment = RAID drive share
   inherit acls = Yes
   path = /local/public
   read only = No
```

How to configure Samba server

The following is the step by step procedure to configure a Samba Server

1. Server IP address is xxx.xxx.0.254 and machine name is Server1

2. Windows machine IP address is xxx.xxx.0.2 and machine name is client2

3. Firewall Should be disabled.

```
#chkconfig iptables off
#service iptables stop
#chkconfig ip6tables off
#service ip6tables stop
```

4. To check if all rpms required for samba are installed or not.

#rpm –qa | grep samba, then it displays the following
Samba-<version-name>
Samba-common-<version-name>
Samba-client-<version- name>
System-config-samba-<version-name>

5. If not, install it using

#rpm –ivh samba

6. Copy the original configuration file as smb.conf.bck

#cp /etc/samba/smb.conf /etc/samba.conf.bck
(This will keep a backup copy of your configuration file.)

7. Now edit the configuration file

#vi /etc/samba/smb.conf

8. To share home directory edit this part of /etc/samba/smb.conf

[homes]
comment=Home directories
browseable =no
writable=yes

9.To share printer edit this part of /etc/samba/smb.conf

[printers]
comment = All printers
path = /var/spool/samba
browseable = no

10.To configure share called IHNCSHARE edit this part of /etc/samba/smb.conf

[IHNC's share]
comment = Testing Samba Server
path = /samba
valid users = user1, user2

11. Save the file with wq command

12. To check your configuration file type testparm

#testparm

13. Create two samba users user1 and user2

useradd user1
useradd user2

14. Create smbpassword for both the users

smbpasswd –a user1
smbpasswd –a user2

15. Now you stop/start the samba services

```
#service smb stop
#service smb start
```

16. To see what is shared from your server through samba for a particular user

```
#smbclient -L Server1 –U user1 or smbclient –L //xxx.xxx.0.1 –U user1
```

17. To see what is shared for a particular host

```
#smbclient –L mac2
```

18. Now, permanently make samba server on

```
#chkconfig smb on
```

Check Your Progress 2

1. List the components required to configure BIND.

……

……

……

……

……

……

……

2. What is Samba Server? Explain its importance.

……

……

……

……

……

……

3.7 SUMMARY

In this unit, installation, configuration and setup of various network services such as Dynamic Host Control Protocol (DHCP), Domain Name System (DNS), Network File System (NFS) and Samba server are explained in detail with examples. This knowledge nay help to understand the concepts and install, configure and commissioning of other network services such as Email, FTP and such related services also. Student has to practice in real time to have more exposure and built confidence in configuration of network services.

3.8 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1. The following are the activities between DHCP Server and DHCP Client.

   - Lease Request: Client broadcasts request to DHCP server with a source address of 0.0.0.0 and a destination address of 255.255.255.255. The request includes the MAC address which is used to direct the reply.
   - IP lease offer: DHCP server replies with an IP address, subnet mask, network gateway, name of the domain, name servers, duration of the lease and the IP address of the DHCP server.
   - Lease Selection: Client receives offer and broadcasts to al DHCP servers that will accept given offer so that other DHCP server need not make an offer.
   - The DHCP server then sends an acknowledgement to the client. The client is configured to use TCP/IP.
   - Lease Renewal: When half of the lease time has expired, the client will issue a new request to the DHCP server.

2. The following are different types of DNS servers:

   Primary (master) name server contains authoritative information about the domains that it serves. In response to queries for information about its domains, this server provides that information marked as being authoritative. The primary is the ultimate source for data about the domain. The secondary name server only carries the same authority in that it has received and loaded a complete set of domain information from the primary.

   Secondary (slave) name server gets all information for the domain from the primary. As is the case for the primary, DNS considers the secondary information about the domain that it serves authoritative.

   Caching name server simply caches the information it receives about the locations of hosts and domains. It holds information that it obtains from other authoritative servers and reuses that information until the information expires.

   Forwarding name server is essentially a caching name server but is useful in cases where computers lie behind a firewall and in which only one computer can make DNS queries outside that firewall on behalf of all the internal computers.

Check Your Progress 2

1. The following components are required to configure BIND.

   Configuration file (/etc/named.conf) is the main DNS server configuration file.

   Zone directory (/var/named) is the directory containing files that keep information about the zone that you create for your DNS server.

   Daemon process (/usr/sbin/named) is the daemon process that listens for DNS requests and responds with information that the named.conf file presents.

Debugging tools (named –checkconf, and named-checkzone) are to determine whetherthe created DNS configuration correctly.

2. Samba is a software package that comes with Red Hat Linux to share file systems and printers on a network with computers that use the session massage block (SMS) protocol. SMB is the protocol that is delivered with windows operating systems for sharing files and printers. In Red Hat Linux, the Samba software package contains a variety of daemon processes, administrative tools, user tools, and configuration files.

   The default Samba configuration file is smb.conf, which is in /etc/samba directory (/etc/samba/smb.conf). If you need to access features that are not available through the samba server configuration file you can edit /etc/samba/smb.conf file as required.

3.9 FURTHER READINGS

1. Computer Networks by Andrew S Tanenbaum , Fifth Edition
2. SA2, Redhat System Administration I & II, Student Workbook
3. Cisco Certified Network Associate Study Guide, Seventh Edition by Todd Lammle
4. Redhat Enterprise Linux System Administration
5. http://en.wikipedia.org/wiki/dhcp
6. http://en.wikipedia.org/wiki/dns
7. http://en.wikipedia.org/wiki/nfs

UNIT 4 NETWORK MANAGEMENT AND SECURITY

Structure **Page Nos.**

4.0 Introduction 70
4.1 Objectives 71
4.2 Networks and Security 71
4.3 User Security Management 72
4.4 Disk Security Management 74
4.5 Security Configuration and Analysis 76
4.6 Account Policies 77
4.7 Permissions and Restrictions 81
4.8 Configuring Network Settings 84
4.9 Advance Troubleshooting 85
4.10 Summary 88
4.11 Answers to Check Your Progress 88
4.12 Further Readings 89

4.0 INTRODUCTION

Network management includes the activities, methods, procedures, and tools that pertain to the operation, administration, maintenance, and provisioning of networked systems. Network Operation deals with keeping the network up and running smoothly. It includes monitoring the network to diagnose and identify problems as soon as possible, ideally before users are get affected. Network administration deals with keeping track of resources or components in the network and how these resources are assigned and do necessary steps to keep the network under control. Network maintenance is concerned with performing repairs and upgrades. For example, when the equipment must be replaced, when a router needs a patch for an operating system image and when a new switch is added to a network and so on. Maintenance also involves corrective and preventive measures to make the managed network run "better", such as adjusting device configuration parameters. Network provisioning is concerned with configuring resources or components in the network to support a given service. For example, this might include setting up the network so that a new customer can receive voice service.

A common way of characterizing network management functions is FCAPS- Fault, Configuration, Accounting, Performance and Security. Functions that are performed as part of network management include controlling, planning, allocating, deploying, coordinating, and monitoring the resources of a network. Network planning, frequency allocation, predetermined traffic routing to support load balancing, cryptographic key distribution authorization, configuration management, fault management, security management, performance management, bandwidth management, route analytics and accounting management are some of the key managerial activities that are to be performed for effective network management and ensure security. A network management system (NMS) is combination of hardware and software used to monitor and administer a computer network or networks.

Data for network management is collected through several mechanisms, including agents installed on network infrastructure, synthetic monitoring that simulates transactions, logs of activity, sniffers and real user monitoring. In the past, network management mainly consists of monitoring whether devices in the network were up or down, but today performance management has become a crucial part of the IT team's role.

As usage of Systems/Applications/ Services through network is increasing day by day, at same time Hackers/Attackers are playing a vital role to destruct the system resources and deface the Websites , etc. Security is essential to protect the resources of systems, services and applications from Hackers or Attackers and in turn protect the sensitive information and data.

Security management is an activity that includes a set of functions that are to be performed to protect telecommunications networks and systems from unauthorized access by persons, acts, or influences. Security functions are for creating, deleting, and controlling security services and mechanisms; distributing security-relevant information; reporting security-relevant events; controlling the distribution of cryptographic keying material; and authorizing subscriber access, rights, and privileges.

4.1 OBJECTIVES

After going through this unit, you will be able to:

- understand the network management and security;
- know how user management can be done in a security perspective;
- understand disk management in a security perspective;
- find account policies and specially password policy;
- find various user permissions and restrictions; and
- understand troubleshooting of network and available tools.

4.2 NETWORKS AND SECURITY

A computer network is a telecommunications network that allows computers to exchange data. The physical connection between networked computing devices is established using either cable media or wireless media. The best-known computer network is the Internet.

Network devices that originate, route and terminate the data are called network nodes. Nodes can include hosts such as servers and personal computers, as well as networking hardware. Computer networks support applications such as access to the World Wide Web, shared use of application and storage servers, printers, and fax machines, and use of email and instant messaging applications.

Networks established either through wired or wireless technologies. Wired networks can be established by using twisted pair, coaxial cable and optical fiber. Wireless networks can be established by using terrestrial microwave communications, communication satellites, cellular and PCS systems, radio and spectrum technologies. All these technologies use different network components that are to be used for routing, switching the data traffic for successful transmission. Some of the routing and switching devices involved in any communication network are the routers, switches, bridges, repeaters, hubs, etc. As many network devices will participate during data transmission from source to destination, ensuring secure communication is a challenging task.

Why do we need Security?

Security is the degree of resistance to, or protection from, harm. It applies to any vulnerable and valuable asset, such as Information Technology infrastructure, a computer network, a person, dwelling, community, nation, or organization. Due to rapid developments in internet technologies, usage of network services for ecommerce applications, banking, education and many such areas are increasing day by day and at same time hackers are also playing a vital role to destruct services and data. Security is essential to protect the network resources and in turn ensure secure data transmission.

Security services

The following are various security services or parameters to enhance the security of a systems, applications, data and are intended to counter security attacks.

- Authentication
- Authorization & Access Control
- Availability
- Confidentiality
- Integrity
- Nonrepudiation

Authentication is the act of establishing or confirming something (or someone) as authentic. It confirms the identity of a person or a system and permits one system to determine the origin of another system. It is essential in online community, where two systems are usually not directly connected

Authorization and Access Control is the level of access control that is permitted to use systems and services

Availability ensures that the system or an Application or a Service is always available to the authorized parties when needed

Confidentiality provides the secrecy of information and allows only authorized users to have access to information.

Integrity ensures that only authorized parties are able to modify computer system assets and transmitted information and provides the correctness of information.

Nonrepudiation ensures that neither the sender nor receiver of a message be able deny the transmission. It is a kind of system that includes authentication, integrity and non-repudiation should be able to detect tampered information and prevent valid information from being falsely rejected.

Deviation in any of the above security services is a breach in security.

4.3 USER SECURITY MANAGEMENT

User Management is an authentication feature that provides administrators with the ability to create users on a system or a service identify and control the state of users logged into the system and even network. User management includes the ability to query and filter users those are currently logged into the system or network, manually log out users, and control user login counts and login times.

Why do you need User Management?

Most security-conscious enterprises today implement some form of authentication and authorization for accessing network resources. In this process, user permissions

can be verified before granting access to resources, and user activity can be monitored through various logging mechanisms. In typical authentication and authorization deployments, administrators have various options available with regard to how users are authenticated, but have little control over how often users are authenticated. User Management enables administrators to control the frequency of user authentication, to ignore cached browser credentials and force the user to re-enter credentials, or to require more frequent authentication only if the user is accessing critical resources. This kind of flexibility allows administrators to implement authentication-based policies that more closely match their network security policies.

The User Management logout capability also provides more secure control over the state of users. For example, when using IP authentication mode, users are identified by the specified IP address until the IP surrogate time expires. If another person were to use that computer before the IP surrogate time expired, they would be treated as the original user. The common solution for preventing this scenario is to decrease the IP surrogate expiry time, causing the user to be challenged more often. Another key benefit of User Management is visibility into active user sessions. Using the Management Console and CLI, administrators can view all active users and filter display data by user, IP address, or realm for easier viewing. This can be useful for identifying the general login status of users or for making real-time decisions such as immediately logging off a user.

How does User Management work?

User Management is based on the concept of users logging in and logging out of a system or a network. A login is the combination of a unique IP address with a unique username in a unique domain. A user is considered logged in when first authenticated to the system or a network. Identifying users as logged in, or active, allows administrators to create flexible User

Management policies to fine tune user access and control.

The majority of User Management is done by framing and introducing a policy. Using policy, administrators can create, delete, modify users and also enforce controls on logging ins, logouts, number of login attempts and other such related.

The following are some of the policies implemented as part of user management in a security perspective:

- Create genuine user-names on a system or network or a service.
- Frequently monitor unauthorized users, if any created, logged in or connected.
- Introduce timestamps on logins and logouts of users and monitor in case of odd timings, if any activities performed.
- Keep tracking on users , who login in long time.
- Clearly add expiry date of a user at the time of user creation itself.
- Enforce policy to deactivate automatically after expiry.
- Introduce multi-level authentication such as authentication with username and password; username, password and IP address; username, password, MAC-address, etc.
- Limit the number of IP addresses associated with a single username.
- Limit the number of logins associated with a single IP address.
- Force a re-authentication to gain access to a particular network resource.
- Limit the login session time allowed in a particular timeframe.

4.4 DISK SECURITY MANAGEMENT

Disk Management is an activity to manage the drives installed in a computer like hard disk drives (internal and external), optical disk drives, and flash drives. Disk Management activities are like partition drives, format drives, assign drive letters, and other such related. For disk management can be done either with help of a tool or with a command to manage system disks, both local and remote.

The following are some of Disk Management functions:

- Create partitions, logical drives, and volumes.
- Delete partitions, logical drives, and volumes.
- Format partitions and volumes.
- Mark partitions as active.
- Assign or modify drive letters for hard disk volumes, removable disk drives, and CD-ROM drives.
- Obtain a quick visual overview of the properties of all disks and volumes in the system.
- Create mounted drives on systems using the NTFS file system.
- Convert basic disks to dynamic disks.
- Convert dynamic to basic disks, although this is a destructive operation.
- On dynamic disks, create a number of specialty volumes including spanned, striped, mirrored, and RAID-5 volumes.

Disk Management

Microsoft Windows utility that was introduced with Windows XP as a replacement to the fdisk command that enables users to view and manage the disk drives installed in their computer and the partitions associated with those drives. Figure 1 shows the snapshot of disk management utility in windows XP. Each drive is displayed followed by the layout, type, file system, status, capacity, free space, % free space, fault tolerance and overhead.

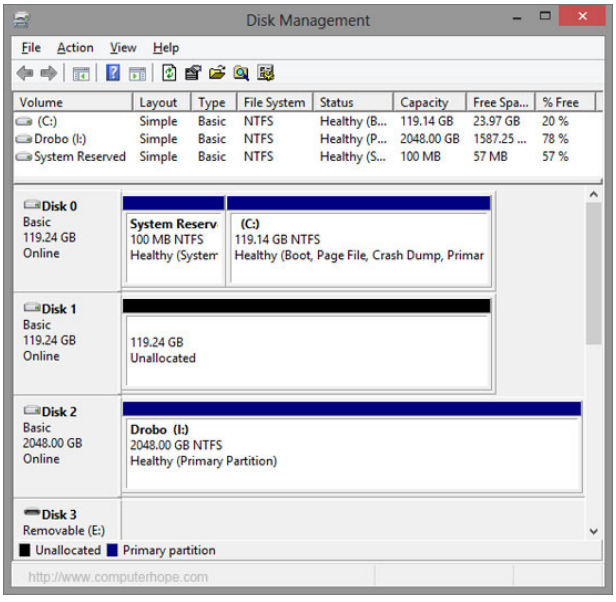


Figure 1: Disk Management in Windows XP

The following is the procedure to open Windows Disk Management

- Click Start, Settings, Control Panel.
- Double-click on Administrative Tools if in Classic View or click Performance and Maintenance and then Administrative Tools if in Category View. Note: If you do not have admin rights to the computer this will not be available.
- Once in the Administrative Tools window double-click Computer Management and then click Disk Management under the Storage section.

Similarly, disk management utilities are available in Linux environment (fdisk command) too to perform various activities as being done in Windows environment.

Disk Management in a security perspective

The following are to be followed for secure disk management:

- Create adequate number of disk partitions
- Allocate adequate storage space in each disk partition as per requirement
- Ensure minimum free space al the times
- Set password for each disk partition or a disk drive
- Scan each disk partition or a disk drive at regular intervals for viruses or worms, etc
- Enforce standard anti-virus software to check and clean viruses in a file before storing into a particular disk partition.
- Update Anti-virus software, periodically
- Disk partitions or disk drives should not be in sharing mode (put in sharing mode only on demand)
- Introduce RAID concept
- Disable disk remote access.
- Apply data encryption at disk storage level too apart from during data transmission
- Implement dish quotas by enforcing upper limits and warning alerts in case reaches to upper limit
- Disk defragmentation to be done, periodically
- Allow disk indexing service

Check Your Progress 1

1. What are various security services or parameters?

……

……

……

……

……

……

2. List the Disk Management functions.

……

……

……

……

……

……

4.5 SECURITY CONFIGURATION AND ANALYSIS

Configuring networked systems and components with adequate security controls in any organization to run or access through network is a critical task. Once, installations, configurations are done, it is very essential to look at the configurations and analyze the setting by keeping security issues in mind before deployment or immediately after deployment.

Security Configuration and Analysis can be done either manually with a check list or with help of a tool that can be used to analyze and configure computer security. Generally, users can use the tool to import one or more saved configurations to a private security database. Importing configurations builds a machine-specific security database that stores a composite configuration. One can apply this composite configuration to the computer and analyze the current system configurations against the stored composite configuration stored in the database.

Best practices for Security Configuration and Analysis

The following are the best security practices to be followed to ensure security in any networked system domain of an organization:

Restrict physical access to computers, especially domain controllers to trusted personnel

Physical access to a server is a high security risk. Physical access to a server by an intruder could result in unauthorized data access or modification as well as installation of hardware or software designed to circumvent security. To maintain a secure environment, you must restrict physical access to all servers and network hardware.

For administrative tasks, use the principle of least privilege

Using the principle of least privilege, Administrators should use an account with restrictive permissions to perform routine, non-administrative tasks and use an account with broader permissions only when performing specific administrative tasks.

Define groups and their membership

At the time of creation of users, the administrator has to define the user under a specific group depending on type of resources or applications where the user needs access.

Secure data on computers

Ensure that the system files and the registry are protected using strong access control lists.

Use strong passwords throughout your organization

Authentication methods require the user to provide a password to prove their identity. These passwords are normally chosen by the user, who may want a simple password that is easily remembered. In most cases, these passwords are weak and may be easily guessed or determined by an intruder. Strong passwords tend to be more difficult for an intruder to detect. Using strong passwords always help to provide an effective defense of resources.

Do not download or run programs that come from untrusted sources

Programs can contain instructions to violate security in a number of ways including data theft, denial of service and data destruction. These malicious programs often masquerade as legitimate software and can be difficult to identify. To avoid these programs, one should only download and run software that is guaranteed authentic and obtained from a trusted source. It is always better to ensure a current virus scanner is installed and functioning in case this type of software does inadvertently wind up on your computer.

Keep virus scanners up to date

Virus scanners frequently identify infected files by scanning for a signature, which is a known component of a previously identified virus. The scanners keep these virus signatures in a signature file, which is usually stored on the local hard disk. Because new viruses are discovered frequently, this file should also be updated frequently for the virus scanner to easily identify all current viruses.

Keep all software patches up to date

Software patches provide solutions to known security issues. Check software provider Web sites periodically to see if there are new patches available for software used in your organization.

Use some available security configuration and analysis tools (for example: secedit.exe is a Windows security configuration editor command tool) to configure, verify and analyze the network settings. This kind of tools can be used for the following:

- For frequent analysis of a large number of computers, as with a domain-based infrastructure,
- To configure security areas not affected by Group Policy settings. It includes the areas such as security on local files and folders, registry keys, and system services. Otherwise, Group Policy settings will override the local settings.
- Do not use such tools, when you are configuring security for a domain or an organizational unit.

4.6 ACCOUNT POLICIES

User accounts or simply login IDs are required to access any system's or network resources, if resources access configured in such a way that they are entry controlled. The system or network administrator has to create usernames on systems depending on need and deed. The administrator has to have a policy on user administration in terms of user account creation, deletion, modification, expire and also have a

password policy in terms of password length, complexity, minimum and maximum password aging, etc.

A user account policy is a document which outlines the requirements for requesting and maintaining an account on computer systems or networks, typically within an organization. It is very important for large sites where users typically have accounts on many systems. Some sites have users read and sign an account policy as part of the account request process.

Introduction to user and Group in Linux

Unix/Linux is a multi user and multi tasking Operating system. Redhat Linux uses User Private Group (UPG) scheme where user always get created with primary group. There is only one primary group per user.

When a user is created in Linux, the following are created

Home directory (home/username)

Mail account (/var/spool/mail/username)

Unique UID & GID

Types of Accounts

Generally, accounts fall into two types such as System accounts(system users) and Normal accounts (normal users). When an user account is created in Linux, the details are stored in the following files:

/etc/passwd
/etc/shadow
/etc/group

where

/etc/passwd contains database of all users created.

Example entry in a /etc/passwd file

u1:x:500:550:prog:/home/u1:/bin/bash

Where
u1 is User ID
X is mask password
500 is UID (user ID)
550 is GID (group ID)
prog is user ID comment field (generally it could be name of user)
/home/u1 is home directory of user ID and

/bin/bash is a shell

The /etc/shadow file contains the encrypted user passwords assigned by the password binary file. Password's are encrypted through DES (Data Encryption Standard) or MD5 (Message Digest Ver.5) Algorithm.

The/etc/group file contains Group name and GID of the group.

Managing Users

A system administration can manage a user's account. The various tasks that a system administrator can perform include adding, modifying and deleting user account.

User account Creation

To create a user, use the useradd command. The syntax is

root# useradd <option> <userID>

-u is UID

-g is primary/group name/GID

-o is : Override

-G is Secondary group

-c is Comment

-d is Home directory

-s: is Shell

Example:

root# useradd murli , it creates user ID murli and assign some default values against UID, GID, and other

user ID can also create by using all options along with the command useradd as specified above.

User ID Modification

To modify a user, use the usermod command. The syntax is:

root# usermod <options> <userID>
The options to usermod command are:
-l is Change user account
-L is to Lock the account
-U is to Unlock the account

Example

Root# usermod murli

User Deletion

To delete a user, use the userdel command. The syntax is

root# userdel <options> <userID>

The option to userdel command is
-r is recursively delete

Example

root# userdel murli

Managing Groups

System administrator can manage a group's account. The various tasks that a system administrator can perform include adding, modifying and deleting group account.

The following are various the commands:

- groupadd command is for group creation
- groupmod command is for group modification
- groupdelcommand is for group deletion

Account Policies

Account policies are required to manage users effectively. Account Policies contains the following:

Password Policy: These policy settings are used for domain or local user accounts. They determine settings for passwords, such as enforcement and lifetimes.

Account Lockout Policy: These policy settings are used for domain or local user accounts. They determine the circumstances and length of time that an account will be locked out of the system.

Account UID policy: Normal user accounts should not have UID (UserID) value 0(zero). Any account has UID value 0 will get root privileges automatically. System administrators should pay attention on UIID values at the time of account creation.

Account GID policy: Normal user accounts should not have GID (User groupID) value 0(zero). Any user account has GID value 0 will get root privileges automatically. System administrators should pay attention on GIID values at the time of user account creation.

Account creation policy: Policy should be implemented on account creation. Account will be created automatically when a user joined in the organization and needs to access on systems.

Account termination policy: Policy should be implemented on account termination. Account will be terminated automatically with or without grace period, when user superannuated or left or expired.

Kerberos Policy: These policy settings are used for domain user accounts. They determine Kerberos-related settings, such as ticket lifetimes and enforcement. Kerberos policy settings do not exist in local computer policy.

Password Policy

Password policy implementation is very essential to secure accounts and at same time secure the systems, the network and the services. The following are some of the parameters to be enforced as part of setting a password policy:

- Enforce password history
- Maximum password age
- Minimum password age
- Minimum password length
- Password strength (must meet complexity requirements)

The Passwords must meet complexity requirements policy setting determines whether passwords must meet a series of guidelines that are considered important for a strong password. Enabling this policy setting requires passwords to meet the following requirements:

- The password is at least six characters long
- Store password using reversible encryption for all users in the domain
- The password contains characters from three of the following four categories
    - English uppercase characters (from A through Z)
    - English lowercase characters (from a through z)
    - Base 10 digits (from 0 through 9)
    - Non-alphanumeric characters (for example: !, $, #, or %)

Account Lockout Policy

Someone who attempts to use more than a few unsuccessful passwords while trying to log on to a system might be a malicious user attempting to determine an account password by trial and error. Account Lockout Policy settings control the threshold for this response and the actions to be taken after the threshold is reached. Generally, the number of login attempts to access any critical systems such as banking or any ecommerce applications systems are not more than three attempts. If any user fails to access such systems 3 times, system will automatically lock such account for a particular period. The lockout period depends on system sensitivity or organization. The following are the parameters to be enforced as part of account lockout policy:

- Account lockout threshold
- Account lockout duration
- Reset account lockout counter after

4.7 PERMISSIONS AND RESTRICTIONS

Privilege is defined as the delegation of authority over a computer system r a network or a service. A privilege is a permission to perform an action on a system or its resources. Examples of various privileges include the ability to create a file in a directory, or to read or delete a file, access a device, or have read or write permission to a socket for communicating over the internet. Users who have been delegated absolute control are called privileged. Users who lack most privileges are defined as unprivileged, regular, or normal users.

On Unix-like systems, the superuser (commonly known as 'root') owns all the privileges. Ordinary users are granted only enough permissions to accomplish their most common tasks.

Unprivileged users usually cannot perform the following tasks:

- Adjust kernel options.
- Modify system files, or files of other users.
- Change the owner of any files.
- Change the runlevel (on systems with System V-style initialization).
- Adjust disk quotas.
- Start or stop daemons.
- Signal processes of other users.
- Create device nodes.
- Create or remove users or groups.
- Mount or unmount volumes
- Execute the contents of any `sbin/` directory

Principle of least privilege

In security perspective, the principle of least privilege also known as the principle of minimal privilege requires that in a particular abstraction layer of a computing environment, every module such as a process, a user or a program depending on the subject must be able to access only the information and resources that are necessary for its legitimate purpose.

The following are the benefits of the principle of least privilege:

Better system stability

For example, applications running with restricted rights will not have access to perform operations that could crash a machine, or adversely affect other applications running on the same system.

Better system security

For example, running a system or a service in standard user mode increases protection against inadvertent system-level damage caused by attacks, malware, such as root kits, spyware, and undetectable viruses.

Ease of deployment

In general, the deployment of any application with fewer privileges is easy.

Superuser

The superuser is a special user account, which has all privileges and is used for system administration. Depending on the operating system, the actual name of this account might be the root, administrator or supervisor. In some cases the actual name is not significant, rather an authorization flag in the user's profile determines if administrative functions can be performed.

In Unix-like computer operating systems, root is the conventional name of the user who has all rights or permissions (to all files and programs) in all modes (single- or multi-user). Regardless of the name, the superuser always has user ID (UID) 0 (zero). The root user can do many things an ordinary user cannot, such as changing the ownership of files or directories and many such others. The superuser account always point at root's home directory.

Filesystem permissions

Most current file systems have methods of assigning permissions or access rights to specific users and groups of users. These systems control the ability of the users to view or make changes to the contents of the filesystem.

Traditional permissions

Permissions on Unix-like systems are managed in three distinct classes such as s user, group, and others. When a file or directory is created, its permissions are restricted by the umask of the process that created it. Files and directories are owned by a user. The owner determines the file's owner class. Distinct permissions apply to the owner. Files and directories are assigned a group, which define the file's group class. Distinct permissions apply to members of the file's group members. The owner may be a member of the file's group. Users who are not the owner or not a member of the group, such users treated under file's others class. Distinct permissions will apply to others.

The effective permissions are determined based on the user's class. For example, the user who is the owner of the file will have the permissions given to the owner class regardless of the permissions assigned to the group class or others class.

Permissions

The following are the three specific permissions (read, write and execute) on Unix-like systems (flavors of UNIX like all Linux versions and others) that apply to each class:

- The *read* permission grants the ability to read a file. When set for a directory, this permission grants the ability to read the names of files in the directory

- The *write* permission grants the ability to modify a file. When set for a directory, this permission grants the ability to modify entries in the directory. This includes creating files, deleting files, and renaming files.

- The *execute* permission grants the ability to execute a file. This permission must be set for executable binaries (for example a compiled C++ program) or shell scripts (for example a Perl program) in order to allow the operating system to run files. When set for a directory, this permission grants the ability to access file contents.

When permission is not set, the rights it would grant are denied. Files created within a directory will not necessarily have the same permissions as that directory.

Changing permission behavior

Unix-like systems typically employ three additional modes. These are actually attributes but are referred to as permissions or modes. These special modes are for a file or directory overall, not by a class.

- The *set user ID* or *setuid*, or SUID mode. When a file with setuid is executed or is on, the resulting process will assume the effective user ID given to the owner class. It enables users to be treated temporarily as root.

- The *set group ID or setgid* or SGID mode. When a file with setgid is executed or on, the resulting process will assume the group ID given to the group class. When setgid is applied to a directory, new files and directories created under that directory will inherit the group from that directory

- The *sticky bit* mode. The typical behavior of the sticky bit on executable files encourages the kernel to retain the resulting process image in memory beyond termination. On a directory, the sticky permission prevents users from renaming, moving or deleting contained files owned by users other than themselves, even if they have write permission to the directory. Only the directory owner and superuser are exempt from this.

These modes are also referred to as *setuid bit*, *setgid bit*, and *sticky bit*, due to the fact that they each occupy only one bit.

Notation of traditional file permissions

Symbolic notation

There are several ways by which permissions are represented. The most common form is symbolic notation as shown by the command `ls -l`.

For example, the following is the output after executing a command ls –l:

[murli@imssit etc]$ ls -l |more

-rw-r--r-- 1 root root 15276 Oct 10 2006 a2ps.cfg

-rw-r--r-- 1 root root 2562 Oct 10 2006 a2ps-site.cfg

drwxr-xr-x 4 root root 4096 Apr 24 2009 acpi

-rw-r----- 1 root root 450 Jan 26 2009 auditd.conf

The first character indicates the type(is file or directory, etc) and is not related to permissions. The remaining nine characters are in three sets, each representing a class of permissions as three characters. The first set represents the *user* class. The second set represents the *group* class. The third set represents the *others* class.

Each of the three characters represent the read, write, and execute permissions:

'r' if reading is permitted, '–' if it is not.
'w' if writing is permitted, '–' if it is not.
'x' if execution is permitted, '–' if it is not

The following are some examples of symbolic notation:

-rwxr-xr-x a regular file whose user class has full permissions and whose group and others classes have only the read and execute permissions.

crw-rw-r-- a character special file whose user and group classes have the read and write permissions and whose others class has only the read permission.

dr-x------ a directory whose user class has read and execute permissions and whose group and others classes have no permissions.

Numeric notation

Another method for representing Unix like permissions is an octal (base-8) notation. This notation consists of at least three digits. Each of the three rightmost digits represents a different component of the permissions: owner, group, and others.

The following shows file permissions in numeric notation (in Octal)

0000 no permissions

0111 execute

0222 write

0333 write and execute

0444 read

0555 read and execute

0666 read and write

0777 read, write and execute

4.8 CONFIGURING NETWORK SETTINGS

Setting up an ideal network in any organization requires the computer systems to host the applications; the network gateway level components such as the routers and switches; the security devices such as the firewall, the intrusion detection or prevention systems (IDS/IPS); the Anti-virus software to protect from viruses; a policy document that facilitates to implement and enforce various policies for effective usage of system resources and network services; the network design document.

For effective implementation of networked systems in any organization, all the required and involved systems or components shall be installed and configured properly. The network services/applications such as Email, DHCP, DNS, NFS, Web servers, etc will have a configuration files with default values. The system or the network administrator has to study each configuration file and set the required values in place of default values at each configuration files. It includes the IPaddresses connected with systems and services, blocking of unnecessary ports, mapping services with allowed ports, netting of public IPs with private IPs in creation of special Demilitarized Zones(DMzs), enforcing proper Access Control Lists at Firewalls, etc. Configuring the network with proper settings in any networked

environment ensures right services at right time always by protecting the environment from hackers or attackers.

4.9 ADVANCE TROUBLESHOOTING

Problem diagnosis and troubleshooting is critical activity which has to be done either manually or automatically (with help of scripts) for effective maintenance of systems and networks and their resources. It can be done locally or remotely.

Problems occur at the network level where systems are connected- due to improper configuration settings done at network components and services; at system level- the physical hardware, the installed operating systems and the system resources.

Basic network issues and troubleshooting tools

Network problems occur due to lack of network connectivity, improper software installations and configurations, insufficient or non-working network hardware, power fluctuations or no power, insufficient or no security devices (firewalls, IDS/IPS- intrusion detection or prevention system) installed in a network, network bandwidth issues, no anti-virus software or not updated anti-virus, and many such related.

Network troubleshooting tools are a necessity for every network administrator. When getting started in the networking field, it is important to have number of tools that can be used to troubleshoot a variety of different network problems and conditions.

Ping

The most commonly used network tool is the ping utility. This utility is used to provide a basic connectivity test between the requesting host and a destination host. This is done by using the Internet Control Message Protocol (ICMP) which has the ability to send an echo packet to a destination host and a mechanism to listen for a response from this host. Simply stated, if the requesting host receives a response from the destination host, this host is reachable. This utility is commonly used to provide a basic picture of where a specific networking problem may exist. For example, if an internet connection is down at an office, the ping utility can be used to figure out whether the problem exists within the office or within the network of the internet provider.

Tracert/traceroute

Once the ping utility has been used to determine basic connectivity, the tracert/traceroute utility can used to determine more specific information about the path to the destination host including the route the packet takes and the response time of these intermediate hosts The tracert utility and traceroute utilities perform the same function but operate on different operating systems, Tracert for Windows machines and traceroute for Linux based machines.

Ipconfig/ifconfig

One of the most important things that must be completed when troubleshooting a networking issue is to find out the specific IP configuration of the variously affected hosts. Sometimes this information is already known when addressing is configured statically, but when a dynamic addressing method is used, the IP address of each host can potentially change often. The utilities are ipconfig on Windows machines and the ifconfig utility on Linux.

Nslookup

Some of the most common networking issues revolve around issues with Dynamic Name System (DNS) address resolution issues. DNS is used by everyone using the internet to resolve commonly known domain names (i.e. google.com) to commonly unknown IP addresses (i.e. 74.125.115.147). When this system does not work, most of the functionality that people are used to goes away, as there is no way to resolve this information. The nslookup utility can be used to lookup the specific IP address(es) associated with a domain name. If this utility is unable to resolve this information, there is a DNS issue. Along with simple lookup, the nslookup utility is able to query specific DNS servers to determine an issue with the default DNS servers configured on a host.

Netstat

Netstat utility is used to display the currently active ports on a Linux machine. This is very important information to find for a variety of reasons. For example, when verifying the status of a listening port on a host or to check and see what remote hosts are connected to a local host on a specific port. It is also possible to use the netstat utility to determine which services on a host that is associated with specific active ports.

Putty

Putty utility is used to connect different systems remotely. Putty is being used to connect to a host via SSH.

Nmap

The nmap utility is one of the most versatile of network tools that is available. Regardless of how much experience a network engineer has, the nmap utility should always be available. Nmap utility can be used for the following:

- port scanning (TCP/UDP)
- version detection
- OS detection
- ping sweeps

Wireshark/tcpdump

Wireshark/tcpdump utilities are the packet scanners that have the ability to capture and analyze individual packets that are sent across a network.

Wireshark includes many different functions that provide the ability to perform a number of different analysis including filtering by conversation (i.e. IPv4, TCP, UDP..) and protocol analysis (HTTP, VoIP protocols (RTP, SIP, H.225..).

Tcpdump is another packet scanner that is available that provides the ability to analyze network traffic and is very easy to configure. Tcpdump is used on a Linux machine (various flavors) and is available for Windows as Windump.

inSSIDer

The inSSIDer utility can be used to not only scan for different networks within the 2.4 and 5 GHz ranges but also list the current signal strengths of different wireless networks within range.

Syslog server

A simple syslog server can be installed in the field to receive network events from key network elements. This information can then be recorded over time and help in determining the cause of a networking problem.

PTRG Network Monitor

The PTRG network monitor utility offers the ability to track the status of different sensors over a period of time. These sensors monitor anything from simple reachability (ping) to the response time of specific services (i.e. HTTP or POP).

System level issues and activities to be done

Generally, system level problems are due to system hardware components, improper operating system installations, improper configuration settings and other such related.

The following are some of the activities to be done by a system administrator:

- Periodically update the Operating system patches, if any
- Remove unnecessary accounts created by default
- Close all ports except the ports required to allow the services
- Harden the operating system
- Regularly monitor the system resources such as the file system, the storage, the log files and analyze.
- Regularly monitor the users and their activities
- Update Anti-virus patches, periodically

Best practices to be followed for ideal networked environment

Organizations require ideal networked systems to be established to run and access various network services or applications. The following points may be kept in mind for the purpose:

- The systems or servers should have adequate resources in terms of processing speed, memory, storage, etc
- Use the Routing and Switching devices with adequate resources
- Configure Firewall Device with proper ACLs(Access Control Lists) to control the network traffic
- Configure IDS/IPS for effective detection of intrusions/intruders
- Configure network configuration files properly for various network services such email, DNS, DHCP etc
- Implement VLANs and enforce traffic limits
- Use appropriate network monitoring tools for effective network monitoring
- Do periodic system and network performance auditing
- Do periodic security audit at system and network level
- Policy to be framed and implemented for effective usage of systems and network resources

Check Your Progress 2

1. Write the standard minimum requirement to create a strong password.

   ……

   ……

   ……

   ……

   ……

2. What is the significance of Nmap utility.

   ……

   ……

   ……

   ……

   ……

4.10 SUMMARY

The unit emphasized on network management and security issues in a network. The user management and disk management in a security perspective are clearly and comprehensively explained. Various user account policies along with the parameters to be set for a password file are explained. The file permissions are explained with different examples. Finally, the need of problem diagnosis and troubleshooting along with available tools was also discussed.

4.11 ANSWERS TO CHECK YOUR PROGRESS

Check Your Progress 1

1. The following are various security services or parameters
   - Authentication
   - Authorization & Access Control
   - Availability
   - Confidentiality
   - Integrity
   - Nonrepudiation
2. The following are some of Disk Management functions:
   - Create partitions, logical drives, and volumes.
   - Delete partitions, logical drives, and volumes.
   - Format partitions and volumes.
   - Mark partitions as active.
   - Assign or modify drive letters for hard disk volumes, removable disk drives, and CD-ROM drives.

- Obtain a quick visual overview of the properties of all disks and volumes in the system.
- Create mounted drives on systems using the NTFS file system.
- Convert basic disks to dynamic disks.
- Convert dynamic to basic disks, although this is a destructive operation.
- On dynamic disks, create a number of specialty volumes including spanned, striped, mirrored, and RAID-5 volumes.

Check Your Progress 2

1. To set strong password, password contains characters from three of the following four categories and minimum 8 characters.

   - English uppercase characters (from A through Z)
   - English lowercase characters (from a through z)
   - Base 10 digits (from 0 through 9)
   - Non-alphanumeric characters (for example: !, $, #, or %)

2. **Nmap** utility is one of the most versatile of network tools and can be used for the following:

   - port scanning (TCP/UDP)
   - version detection
   - OS detection
   - ping sweeps

4.12 FURTHER READINGS

1. Computer Networks by Andrew S Tanenbaum , Fifth Edition
2. SA2, Redhat System Administration I & II, Student Workbook
3. Cisco Certified Network Associate Study Guide, Seventh Edition by Todd Lammle
4. Redhat Enterprise Linux System Administration
5. Fundamentals of network security by Eric Maiwald, Dreamtech publisher
6. Linux system security, the Administrators guide to open source security tools by Scott Mann. Ellen L. Mitchell, Second edition, Pearson Education.
7. http://en.wikipedia.org/wiki/security